MN804

mike 2 manual

martin research
3336 Commercial Avenue
Northbrook, IL. 60062
(312) 498-5060

Preserved by
aconit.org
aconit

modular micros | martin research

Dear Customer:

We hope that this *modular micro* Manual is useful to you.

In order to keep this Manual current, Martin Research maintains a special Manual update mailing list. Already on this list are those customers who purchased this Manual, or a *modular micro* system, directly from Martin Research. Initially, all such customers automatically receive a one-year's subscription to these updates, ending December, 1976. Until then, supplements are being sent without charge to fill in incomplete sections in the Manual.

If you wish to confirm your subscription to the Manual, please use the coupon provided below. Also, please keep us current on your address. Martin Research cannot be responsible for delivery of updates unless notified of any address change, preferably one month in advance. Write to the attention of the MANUAL UPDATE DEPARTMENT.

Send to: Martin Research, 3336 Commercial Ave., Northbrook, Illinois 60062 USA

ATTN: MANUAL UPDATE DEPARTMENT

_____ Please make sure my subscription is entered.

_____ Please enter a subscription; *modular micro* purchased from_____.

_____ Change of address, effective on_____.

NAME_____

ADDRESS_____

_____

_____ZIP CODE_____

REV. A

Martin Research hereby solicits your comments and criticisms regarding this Manual. Users' ideas for hardware and software for the *modular micro* series will gladly be received. Subject to evaluation and editing, those judged most useful will be distributed to other Manual holders.

This sheet may be used for your comments, and returned to Martin Research.

© 1975, Martin Research, Northbrook, IL 60062 USA. All rights reserved.

5	4	3	2	Rev	ECO	By	Date
A	A	A	A	A	016	KB	12/4/75

REV. A

SEC. 1 CHECKLIST OF CONTENTS

This edition of the Manual includes the data sheets (prefix *DS*), operating instructions (*OP*), service notes (*SV*), software packages (*SP*), and other materials which are listed on this page.

SECTION	SECTION TITLE	FORM NUMBER	CONTENTS
1.	Modular Micros: An Introduction	DS800	*(This data sheet.)*
2.	Construction: Notes for the Kit-Builder	KN800	Construction notes
3.	Microcomputers: An Introduction	DS801	Intro to microcomputers
4.	The Modular Microcomputer System	DS804	The *MIKE 2* system
5.	Operating Instructions	OP804	Operating the *MIKE 2*
6.	The Central Processing Unit (CPU)	DS401	CPU board
		DS441	CPU board
		DS2202/3	8008 data sheet
		DS9000	Short Stack option
		DS9007	16-Channel Scope Display
7.	Memory: ROM, RAM	DS403	PROM/RAM board
		DS405	4K RAM board
		DS423	PROM/RAM board
8.	Input and Output Devices	DS415	Breadboard
		DS420	Console board
		DS425	Breadboard
		DS9001	Universal I/O board
9.	Power Supply		
10.	Accessories and Peripherals		
11.	Maintenance, Troubleshooting, Service, and Warranty Information	SF9	Replacement parts list
		SF8	Warranty; returns
		SV423	Service notes, 423 board
12.	Resident Software and User's Programs	SP2202	8008 instructions
		SP2210-1	MONITOR 8
		SP2210-4	RAM test program
		SP2210-13	MIKE 2 Monitor

modular micros!

We are proud to announce the *modular micros* from Martin Research. These versatile printed circuit modules can be configured to produce a variety of microcomputer systems, from the simple to the complex.

BEST VALUE !

Not just a collection of parts, with the wirewrapping left to you . . . . but complete with commercial-grade printed circuit boards, made by top PC houses, with plated-through holes. Minimizes construction time, and greatly reduces troubleshooting.

MOST ADAPTABLE!

Our universal bus structure is compatible with all standard eight-bit microprocessors! This is a major advantage for the user who wishes to upgrade his system in the future. And, it is ideal for the OEM building prototypes with competing processors. Memory and accessory boards remain compatible when the CPU is changed. The *MIKE 2* system, now being shipped, is based on the popular 8008 processor, and the 8080-based *MIKE 3* is in production (available November, 1975). Other CPUs are planned.

MOST FLEXIBLE!

The system architecture, designed for optimum flexibility, allows any board to be inserted in any position on the bus. A flat cable and snap-on fifty-pin connectors make for easy expansion.

SUPERIOR SYSTEMS DESIGN!

Programs are entered on a calculator-style keyboard, and six bright LED digits display data and memory addresses. A Monitor program in a PROM makes program entry easy. This is a significant advance over console designs with cumbersome arrays of toggle switches and lamps.

FAST MEMORY AVAILABLE!

Our 4K static RAM board uses RAM chips fast enough--450 ns--to avoid making the CPU wait for memory access in most applications. Now in stock, this board is price-competitive with others' memory boards with far slower specifications. Warning: some manufacturers using slow RAM do not inform the buyer of this fact. The result is that the CPU must wait for memory, resulting in a computer much slower than the specs might imply.

BEST FOR YOUR APPLICATION!

With microcomputer systems starting at under $400, the *modular micros* are ideal for:

- The computer enthusiast, in his home;
- The professional engineer or programmer, to keep up with the state of the art;
- The manufacturer, for prototyping microcomputer systems, or for incorporation in his product.

(OEMs: write for further information.)

REV. A

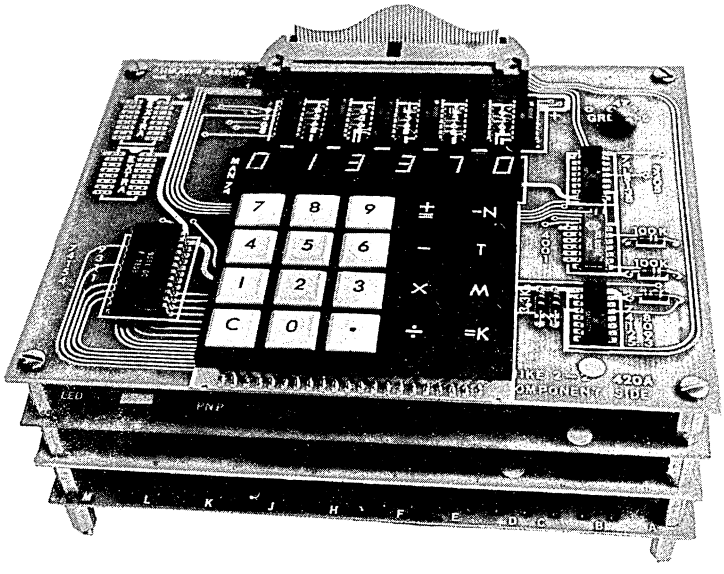
mike 2

The basic *MIKE 2* system includes a 441 CPU Board, with an 8008 microprocessor; a 420 Console Board, with a 20-key keyboard and six fully-decoded digits; a 425 Breadboard, for assembling custom circuitry; and a 423 PROM/RAM board, with 256 bytes of RAM, and a PROM containing a versatile Monitor program. Fully assembled and tested, this *MIKE 2* system is available as the AT 804. Each board is described more fully in the pages that follow.

With the basic *MIKE 2* system, the user can store programs in RAM and execute them, using the calculator-type keyboard. Instructions are entered in octal form, and are displayed automatically on the six bright digits. The user can rapidly address any location in memory, and display the contents at that address. The Monitor automatically alternates between memory address and contents at the selected location. All these functions, supported by the Monitor, are features of even the smallest *MIKE 2* system.

For expansion, the user could add a Short Stack option and another page (256 bytes) of RAM. Now, when a program is interrupted, all registers are saved, so that the original program resumes accurately when the interrupt is completed. And, program breakpoints can be inserted, so that the contents of registers and status flags can be examined at the breakpoint locations. A logical adjunct would be the Demo PROM, supporting memory test routines, a digital clock program, and other useful functions.

For a still more versatile system, the user might install seven PROMs programmed by Martin Research with the MIL MON-8 program. With a Teletype interface board, the MON-8 allows the user to type in his program symbolically (eg, OUT 015 for an Output 15 instruction). The MON-8 also allows for symbolic or numeric program listings and many other functions.



The *MIKE 3* system--based on the powerful 8080--is compatible with the *MIKE 2* bus structure. Available in November, the *MIKE 3* is the next logical step in upgrading the microcomputer.

To stay informed on these exciting modular micro *developments, keep in touch with Martin Research!*

Circuit boards and systems in the *modular micro* series which are sold in assembled and tested form (prefix *AT*) are shipped ready for operation. Purchasing tested boards is the best way to obtain a debugged system with a minimum of labor and testing.

Modular micro boards and systems are also sold as kits (prefix *KT*). To date kits have usually been shipped partially assembled. Beginning with the introduction of the 804 system, all *modular micro* boards are sold with sockets for every IC included. The exception is the Model 420 Console board, where for appearance's sake, the ICs are soldered directly into the board.

When a *modular micro* board is shipped as a collection of parts plus PC board, supplemental instructions are provided.

The remainder of this sheet describes certain construction details common to components in the *modular micro* series.

SEC. 1 WARNING! STATIC DESTROYS MOS!

Some critical microcomputer components are manufactured using *MOS* (metal oxide semiconductor) technology. These devices can be destroyed by static electricity. Examples include the 8080, 6800, and 8008 microprocessors; 1702 PROMs; and 2102 and 2112 RAMs.

A MOS device should either be stored in black conductive foam or a metal IC container, or be plugged into an IC socket which has been soldered into a *modular micro* board. Avoid unnecessary handling of unprotected MOS ICs. Use their ceramic or plastic packages to pick them up, rather than their metal terminals. Avoid wearing nylon clothing, or walking across carpeted rooms, especially under dry atmospheric conditions. Either work on a grounded bench, or contact a grounded surface (water pipe or electrical conduit) by hand to discharge static buildup *before* handling MOS devices. Remove power from a circuit before inserting or removing a MOS IC.

SEC. 2 ASSEMBLY OF THE *MODULAR MICRO* STACK

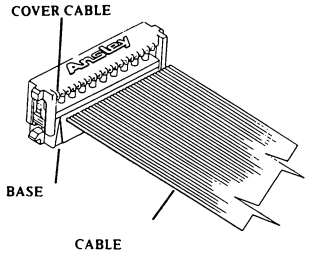
The standard *modular micro* board measures 5.5 by 7.0 inches (140 mm by 178 mm), and is punched with holes at the four corners to clear #6-32 threads. The boards are assembled as a stack, with the boards separated by 5/8-inch hexagonal spacers. Each spacer has one female and one male end, with #6-32 threads; spacers may conveniently be tightened by hand or with a ¼-inch hex wrench.

Each board is equipped with a fifty-pin PC board *header,* plugging into a *transition connector* on a fifty-wire cable. Boards may be added to the system by attaching transition connectors to the cable, mounting additional hex spacers on the stack, and plugging in the new components. During testing and prototyping, only one or two spacers are necessary to separate boards--rather than using all four columns. This facilitates rapid insertion and removal of boards from the stack.

SEC. 3 FIFTY-WIRE CABLE SYSTEM

The fifty-pin connection system used in the *modular micro* series offers maximum flexibility to the user. Enough cable is provided with each system for a reasonable amount of expansion; or, cable may be purchased from Martin Research. When the boards are to be assembled in a stack, using 5/8" spacers, the recommended distance (center to center) between transition connectors is 3/4" (1.9 cm). Longer spacing may require the cable to bend too sharply, putting strain on the connections to the boards.

The user can add a transition connector to the cable with an ordinary bench vise. The connectors supplied by Martin Research are recommended, since some brands require special assembly equipment, and once attached cannot be removed. The transition connector comes in two parts, a *cable cover* and *base.* Assembly steps: with the cable cover placed loosely over the base, slide the cable into its desired position. Now hold the cable assembly up to a light and make sure each of the fifty wires is directly above one of the fifty V-notched teeth on the connector base. Ensure also that the transition connector is at right angles to the run of the cable. Press the cable cover and base together by hand, clamping the cable. Using a vise, *gently and evenly* apply pressure to lock the teeth into the wire. Do this in gradual steps until the cable cover locks onto the base. The ears at the ends of the cable cover might not automatically lock onto the tabs on the base, even when the connector is securely clamped. Try locking the two connector sections by hand before applying still more vise pressure, which could damage the connector.



The Ansley brand of transition connectors may be removed from the cable and repositioned. If the work is done carefully, both connector and cable will still be serviceable.

©1975, Martin Research. Pp. 1-2, Rev. A, Nov. 1975. All rights reserved.

This Manual includes complete circuit descriptions for all microcomputer components, with circuit diagrams and software listings. As such, it provides a good introduction to microprocessor technology.

For the reader who is new to microprocessors, a recommended start is to review the manufacturer's data sheet on the microprocessor, reprinted in the *CPU* section of the Manual. Then, read the circuit description of the CPU board. Finally, look over the chart on CPU instructions in the *Software* section.

The microcomputer designer wishing a more comprehensive treatment may wish to study the book published by Martin Research, *Microcomputer Design.*

The reader who comes to electronics for the first time may need a more general introduction. This will be provided by supplements to this section of the Manual, to be transmitted to Manual holders at a later date.

In all cases, the best introduction is hands-on experience with a functioning microcomputer. This is best provided by working with a *modular micro* system, understanding its operations, and gaining facility in programming.

The basic *MIKE 2* system includes four circuit boards. The Model 804 includes a 441 CPU board, a 420 Console board, a 425 Breadboard, and a 423 PROM/RAM board. (The earlier Model 803 includes a 401 instead of a 441, a 415 instead of a 425, and a 403 instead of a 423. The circuitry is substantially the same.) Complete details on each of these components appear in the *MIKE 2* Manual.

Based on the 8008 microprocessor, the *MIKE 2* may easily be converted for operation with the 8008-1, a premium device operating 60% faster than the standard 8008. More important, the *MIKE 2* uses a flexible bus structure which makes the hardware upward-compatible with systems based on other standard eight-bit processors. These include the *MIKE 3*, based on the 8080, now available, and the 6800, 6501, and 6502 CPUs, to be announced.

SPECIFICATIONS

Processor	Intel 8008-1 PMOS 8-bit microprocessor. Eight bits per byte, instructions one to three bytes in length. Clock frequency, 500 KHz 8008, 800 KHz 8008-1. Register-to-register add time, 20 us, 12.5 us 8008-1. 48 standard 8008 instructions. CPU includes 8 14-bit address registers, 8 8-bit index registers, and seven-level stack.
Memory	8008 address capacity, 16K words. Basic system comes with 256 bytes of RAM, 256 bytes of PROM (the *MIKE 2* Monitor program). May be expanded with boards provided to 1K of RAM, 2K of PROM. Accessory 4K static RAM board is available. Speed of all memory provided adequate for 8008-1 operation without wait cycles.
Input/Output	Provided with Console board, functioning as three output ports (for display) and one input port (for keyboard). Inputs and outputs may easily be added using a single-chip strobe decoder and standard TTL latches or bus drivers. Details are provided in the *MIKE 2* Manual. Up to 32 I/O port addresses are available; may be expanded by conditional input addressing techniques. Optional I/O available: interfaces to printers, keyboards; others in development.
Software	Provided with versatile *MIKE 2* Monitor, supporting machine language programming in octal code through the Console board. A complete operating system, MONITOR 8, is optional.
Packaging	Four boards, 5.5 by 7.0 inches, separated by standoffs and connected by 50-pin ribbon cable. System may be expanded to ten or more modules by adding boards to the stack.

SPECIFICATIONS (cont'd)

Power — Power supply optional. Voltages and currents required listed on module data sheets.

TECHNICAL DATA

SEC. 1 INTRODUCTION

Operating details on the major components of the *MIKE 2* are provided in separate data sheets. These appear in the following sections of the *MIKE 2* Manual: Part 6, CPU; Part 7, Memory; Part 8, I/O; and Part 12, Software.

The following discussion concentrates on *MIKE 2* characteristics at a system level. For more detailed explanations on the circuitry and programming, see the appropriate data sheets in the Manual.

SEC. 2 BUS STRUCTURE

2.1 The *modular micro* bus is based on a fifty-pin connector system. Bus architecture is compatible with many standard eight-bit microprocessors. Figure 2.1.1 shows the signals and power supply voltages assigned to each pin on the bus.

The first eight pins are the bidirectional data bus, on which data normally flows between the CPU and memory, I/O devices, and peripherals. The next sixteen pins are the address bus, used to address memory (and I/O devices). The 8008 system uses only fourteen terminals on the address bus for memory addressing, since it interfaces with a maximum of 16K bytes. The extra two lines are provided for processors which address up to 64K bytes of memory.

Pins 25 through 38 are control signals, used to control data flow within the microcomputer. The functions of these signals are indicated in the chart; detailed descriptions appear in various data sheets, especially that on the CPU board. Note that there are several minor differences between the early 8008 CPU board (Model 401) and the revised version (Model 441): the 401 uses four terminals which are redesignated for later boards. These are pins 25, 26, 27, and 37. Circuitry on the 401 board which is connected to these pins may need to be disconnected from the bus if the system is later upgraded. The 441 board is compatible with the 8080 and later processor systems. Shown in the third column is the 471 board, an 8080-based CPU; only a few signals exist which are not present in the 8008 processor. Note that Martin Research provides the $\overline{\text{IOW}}$ and $\overline{\text{IOR}}$ signals only as a convenience, but does not use them in standard 8080 systems, enhancing compatibility with the 8008 system.

© 1975 Martin Research, Northbrook, IL 60062 USA. All rights reserved.

4	3	2	Rev.	ECO	By	Ck	Date
A	A	A	A	016	BR	KB	12.4.75

REV. A

	NAME OF SIGNAL CPU Board Used						
PIN #	401	441	471			Direction	Function/Comments
1	DB7					In/Out	Data bus, high-order bit
2	DB6					In/Out	
3	DB5					In/Out	
4	DB4					In/Out	
5	DB3					In/Out	
6	DB2					In/Out	
7	DB1					In/Out	
8	DB0					In/Out	Data bus, low-order bit
9	DH7					Out	High-order address, high bit
10	DH6					Out	DH7=DH6=0 in 8008
11	DH5					Out	
12	DH4					Out	
13	DH3					Out	
14	DH2					Out	
15	DH1					Out	
16	DH0					Out	High-order address, low bit
17	DL7					Out	Low-order address, high bit
18	DL6					Out	
19	DL5					Out	
20	DL4					Out	
21	DL3					Out	
22	DL2					Out	
23	DL1					Out	
24	DL0					Out	Low-order address, low bit
25	Ø1	—				Out	
26	Ø2	$\overline{ADV}$				Out	$\overline{ADV}$= address valid
27	SY	—	$\overline{IOR}$			Out	$\overline{IOR}$= I/O read
28	$\overline{DIN}$					Out	Universal I/O strobe
29	$\overline{PCW}$					Out	Memory write cycle
30	$\overline{MEMR}$					Out	Memory read signal
31	—	—	$\overline{IOW}$			Out	$\overline{IOW}$= I/O write
32	$\overline{BSY}$					In	$\overline{BSY}$=RDY= wait request
33	INTS					In	Interrupt disable
34	INTR					In	Interrupt request
35	$\overline{STEW}$					Out	$\overline{MEMW}$= memory write signal
36	$\overline{MEM}$					Out	Memory read or write
37	$\overline{ST4}$	—	$\overline{HOLD}$			In ($\overline{HOLD}$)	$\overline{HOLD}$= CPU disable request
38	—	—	$\overline{HACK}$			Out	$\overline{HACK}$= hold acknowledge
39-42	GROUND						
43-44	+12 V						
45	-12 V						
46-47	-9 V						
48-50	+5 V						

SEC. 2 BUS STRUCTURE (cont'd)

Power supply connections are made via pins 39 through 50. A power supply board may be outfitted with 50-pin connectors and plugged into the *modular micro* bus, or may be wired into the system on the Breadboard, using the holes provided for this purpose.

SEC. 3 MEMORY AND I/O ADDRESSES

3.1 *Memory Addresses* There are only a few fixed memory assignments in the *MIKE 2* microcomputer. The MIKE 2 Monitor occupies page 000 in memory; user's RAM resides in page 010; and supplemental RAM necessary for interrupt handling resides in page 013. All other memory pages may be assigned by the user at his convenience. Further details on memory addressing appear in the data sheet on the PROM/RAM board, in the *Memory* section of the Manual.

3.2 *I/O Addresses* The *MIKE 2* Console board uses output ports 15, 16, and 17 for addressing the six digits, and input 6 for scanning the Console keyboard. Input 5 is assigned to the Short Stack option (see DS9000 in the *CPU* section of the Manual). If the MONITOR 8 software package is installed, Input 0 and Outputs 12 and 13 are assigned for interfacing to a teleprinter. All other I/O assignments are at the user's option. For details on I/O interfacing, see the data sheet on the Universal I/O Kit, KT9001, in the *I/O* section of the Manual.

REV. A

The *MIKE 2* microcomputer is supplied with a Console board, featuring a calculator-style keyboard with twenty keys, and a six-digit display. The keyboard is used to program the microcomputer, and the display is used to visualize the results.

The keyboard's twenty keys include the digits *0* through *7* and a number of special-purpose keys. Each of these special keys is assigned a function useful in programming the microcomputer. A table of these functions appears on the following page.

In order to introduce the user to the *MIKE 2*, we will introduce the keys on the keyboard, one by one. We start by assuming that the assembled *MIKE 2* is connected to a suitable power supply; the power is turned on; the six-digit display glows red, probably displaying some random collection of the digits *0* through *9* and *A* through *F;* and the user is sitting in front of the micro with a puzzled expression.

> *Some of the more technical details on microcomputer operations will be described in indented paragraphs in italic type, like this. The beginner may choose to skip these sections for now.*

SEC. 1 STARTING THE MICRO WITH THE INTERRUPT KEYS

In order to start up the microcomputer, it is necessary to provide it with an *interrupt.* Two keys on the Model 420 Console Board are associated with the interrupt function: *T* and *-N.* In order to interrupt the system, it is necessary to press first the *PRESET INTERRUPT (T)* key, and then the *INTERRUPT (-N)* key. The reader should try this.

The display now begins to alternate between two messages:

013370 — CCCxyz — 013370 — CCCxyz (etc.)

where *xyz* is a set of three digits with random values between 0 and 7. Once the display begins to blink in this manner, the microcomputer has been started and is ready to be programmed.

> *The PRESET INTERRUPT (T) key sets a flip-flop on the 420 Console Board, which causes the Interrupt Request (INTR) line on the microcomputer bus to go low. The INTERRUPT (-N) key resets this flip-flop, causing the INTR line to go high again. As this line goes high, the 8008 microprocessor recognizes an interrupt request. Circuitry on the CPU board automatically sets the DH and DL registers to 000 000, pointing to the first location in the 1702A PROM on the PROM/RAM board. This gives control to the M2M MONITOR program, which takes over.*
>
> *The number 013 370 is an address in RAM memory; xyz is the contents of memory at that address. The full significance of these numbers is explained in Section 5 below.*

REV. A

SEC. 2 SETTING UP AN ADDRESS TO STORE A PROGRAM IN MEMORY

In order to make the microcomputer execute a program, the user must generally write a series of instructions into memory, and then execute them. The first step is to choose a suitable block of random access memory (RAM) into which to write the program. In the *MIKE 2* system, without any memory beyond that supplied, there are 256 (decimal) bytes (eight-bit words) of RAM, starting at location 010 000, and continuing through location 010 377. In preparation for storing a program, we will make the microcomputer point to location 010 000.

We assume that the microcomputer has received its initial interrupt and is still blinking the messages *013370* and *CCCxyz.*

First, set up the high-order memory address, that is, *010.* Punch *1, 0.* (It is not necessary to enter leading zeroes.) The display will read *AAA010.* Punch the *LOAD H (X)* key. Now the display reads *EEE000.*

> *The MONITOR program takes care of shifting digits to the left as new digits are entered, as on a handheld calculator. At the same time that a digit is being entered into the display, the value is also being stored in the 8008's internal E register. When the LOAD H key is pressed, the MONITOR transfers the contents of the E register to the H. Then, the contents of the E register are cleared to 000, and the display reads EEE000.*

Second, set up the low-order memory address, that is, *000.* Since the display already reads [*EEE*]*000,* it is unnecessary to enter the *000.* Simply punch the *LOAD L* (÷) key.

> *The E register was cleared to 000 at the end of the preceding LOAD H operation. The MONITOR loads the 8008 L register from the E.*

Now the display will begin to alternate between two messages:

010000 — CCCxyz — 010000 — CCCxyz (etc.)

where *xyz* once again is some random number. The meaning: we are looking at location 010 000 in memory; its present contents are *xyz.* This is the first location in the *MIKE 2's* array of RAM memory, and we are now ready to load in a program.

SEC. 3 A SHORT TEST PROGRAM

Here is a short program for testing operation of the *MIKE 2*. First, the program; then, how you load it in and execute it.

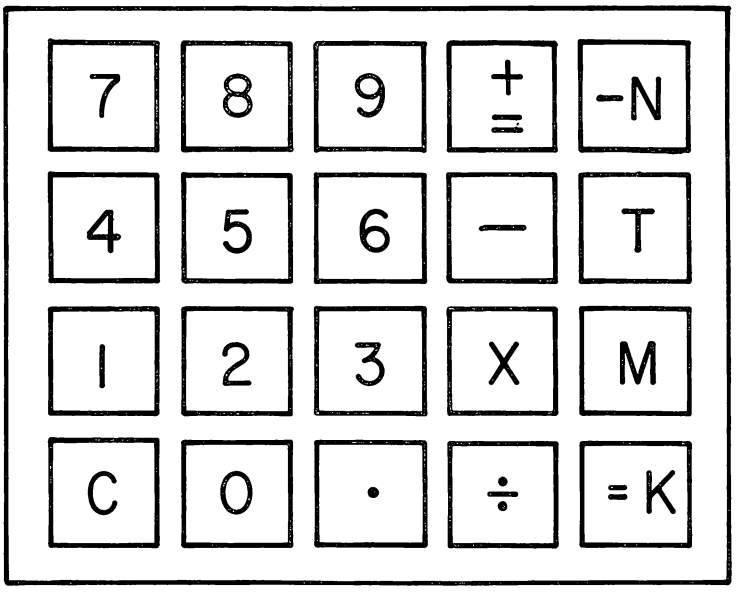
© 1975 Martin Research, Northbrook, IL 60062 USA. All rights reserved.

7	6	5	4	3	2	Rev	ECO	By	Ck	Date
A	A	A	A	A	A	A	016	KB	BR	12/4/75

REV
A

KEYBOARD FUNCTIONS

420 CONSOLE BOARD
MIKE 2 SYSTEM



SYMBOL	HEX	FUNCTION
+ =	A	WRITE
—	B	READ
X	C	LOAD H
• ÷ •	D	LOAD L
•	E	EXECUTE
C	F	CONTINUE
-N	—	INTERRUPT
T	—	PRESET INTERRUPT
M	—	CONTROL 1
=K	—	CONTROL 2

NOTE: PROPRIETARY INFORMATION. NOT TO BE DISCLOSED WITHOUT EXPRESS WRITTEN PERMISSION.

DRAWN BY Schneider DATE 11-18-75
CHECKED BY DM DATE 11-18-75
DRAWING NO. OP 804 (FIG 1)

EF4A

modular micros martin research

SEC. 3 A SHORT TEST PROGRAM (cont'd)

LOCATION	*OCTAL*	*MNEMONIC*	*COMMENTS*
010 000	304	LAE	LOAD A REGISTER FROM E REGISTER
010 001	004 021	ADI 021	ADD 021, I.E., 00 010 001 IN BINARY, OR, REGROUPING, 0001 0001.
010 003	100 007 010	JFC 010 007	IF THE A REGISTER HAS NOT YET OVERFLOWED, FALL THRU THE NEXT INSTRUCTION OVER-
010 006	250	XRA	FLOW; CLEAR A REGISTER
010 007	340	LEA	STORE THE COUNT IN THE E REGISTER
010 010	133	OUT 15	OUTPUT TO DIGITS 1 & 2, 4 BITS FOR EACH DIGIT
010 011	135	OUT 16	OUTPUT TO DIGITS 3 & 4
010 012	137	OUT 17	OUTPUT TO DIGITS 5 & 6
010 013	045	RST 040	JUMP TO LOCATION IN M2M PROGRAM WHICH DEVELOPS 1.2 SECOND DELAY
010 014	104 000 010	JMP 010 000	START AGAIN

The steps for loading the above program are listed below.

1. Punch *3, 0, 4.* (Display reads *AAA304*).
2. Punch the *WRITE* (±) key.

   This key causes the 8008 to write the value contained in its E register, into memory, at the location defined by the H and L registers--i.e., 010 000.

   The display should now alternate between the readings

   010001 — CCCxyz — 010001 — CCCxyz (etc.)

Notice that, after the *WRITE* key is pressed, the MONITOR program automatically proceeds to the next memory location.

3. Punch *4.* (Display reads *AAA004*). Now punch the *WRITE* key. This loads the instruction *004* into location 010 001, and the MONITOR now steps on to the next location in memory, 010 002.

SEC. 3 SHORT TEST PROGRAM (cont'd)

4. *21, WRITE.*
5. *100, WRITE.*
6. *7, WRITE.*
7. *10, WRITE.*
8. *250, WRITE.*
9. *340, WRITE.*
10. *133, WRITE.*
11. *135, WRITE.*
12. *137, WRITE.*
13. *45, WRITE.*
14. *104, WRITE.*
15. *WRITE.* (Unnecessary to enter in *000*).
16. *10, WRITE.*

17. At this point the whole program should have been loaded properly into memory. Let's check. Punch the *LOAD L* (÷) key once again to return the memory address to 010 000. The display should now alternate between 010000 and CCC304--the *304* being the instruction we entered in Step 1, above.

18. Now push the *READ* (-) key. The display should now move to the next location in memory and alternate between 010001 and CCC004. This was the second instruction entered. Just as with the *WRITE* key, the *READ* key automatically goes to the next location in memory--but without changing what is stored there.

19. Keep pressing the *READ* (-) key, checking the contents of memory through location 010 016.

20. If there is a mistake, enter the last three digits of the memory address at which the mistake occurred, using the number keys. Now push the *LOAD L* (÷) key to set the memory pointer back to that location. Enter the correct instruction using the number keys, and push the *WRITE* (±) key to write the correct value into memory.

21. Now execute the program, using the *EXECUTE* key (·). The display should proceed to count through the sixteen hexadecimal numbers, all digits at once, changing about once every 1.2 seconds. (An 8008-1 system will count about once every 0.75 second.)

    111111 – 222222 – 333333 – 444444 – 555555 – 666666 – 777777 – 888888 – 999999 – AAAAAA – bbbbbb – CCCCCC – dddddd – EEEEEE – FFFFFF – 000000 – 111111 – 222222 (etc.)

SEC. 4 *8* AND *9* KEY FUNCTIONS

The keys *8* and *9* are assigned special functions by the MIKE 2 Monitor. Since the keyboard is read in octal format, only the digits *0* through *7* are normally used in entering data or memory addresses into the computer. The Monitor interprets the *8* as an octal *010*--which provides a shorthand method of entering this octal value. Recall that location 010 000 is where many user programs are stored.

The *9* key causes the microcomputer to execute a JUMP instruction to memory location 001 000. This is the first location of the second page in memory, a convenient place for the user

SEC. 4 *8* AND *9* KEY FUNCTIONS (cont'd)

to install a second PROM. For example, the optional *MIKE 2* Demo PROM is usually inserted here. If no PROM has been installed at this location, the computer will usually hit a HALT instruction when the *9* key is pressed, and must be interrupted to resume operations.

SEC. 5 CONTINUE KEY: HANDLING INTERRUPTS

The interrupt feature of the microprocessor is used not only to start the computer, but to *jam* a high-priority program routine into the middle of the main line program. After the interrupt routine is over, the main line program *continues* where it left off.

A *MIKE 2* requires certain hardware to *save* and *restore* the 8008's internal registers and flags during interrupts. This is described in the data sheet for the Short Stack option in the *CPU* section of the *MIKE 2* Manual.

Assuming that the Short Stack has been installed (along with page 013 in RAM), when the *MIKE 2* receives an interrupt, the Monitor program will save register and flag status in RAM memory as follows:

A register:	location 013370.	E:	013374.
B:	013371.	Flags:	013375.
C:	013372.	H:	013376.
D:	013373.	L:	013377.

The user can try this out by executing the sample program in Sec. 3 above, and interrupting the microcomputer while the program is operating (press *T, -N*). The digits on the Console board will now start blinking at 013370, just as they do when the computer is first started up (Sec. 1). The contents of the A register, at the time when the interrupt request was received by the 8008, are located here. By successively pushing the READ (-) key, the user can inspect the other locations, through 013377.

The flag bits which are stored at location 013375 include Carry (C), Zero (Z), Parity (P), and Sign (S). The 8008 data sheet explains how these are set by the ALU, rotate, and increment/decrement instructions. At location 013375, the flags are stored in the 8-bit memory word as follows:

D7	D6	D5	D4	D3	D2	D1	DØ
C	S	S	Z	Z	Ø	Ø	P

After having interrupted the test program and inspected saved status information, now press the CONTINUE (*C*) key. The counting program should resume where it left off.

SEC. 5 CONTINUE KEY: HANDLING INTERRUPTS (cont'd)

Note that if the test program is interrupted *twice* before the CONTINUE key is pressed, the sequence is lost and the microcomputer halts. It is unresponsive to programming, and the red light on the CPU board goes on. It is necessary to interrupt the CPU again, to start the microcomputer.

What happened? The 8008 program counter stack has overflowed. The Monitor program not only saves registers and flags upon receipt of an interrupt; it also must save the *memory address* from which the main line program was fetching instructions at the time when it was interrupted. This instruction address saving is done by using the 8008's CALL and RETURN instructions, which, in conjunction with the 8008's PC register stack, save instruction addresses in the 8008's internal PC register stack. This stack has only seven levels, and some of these are used by *MIKE 2* Monitor program subroutines, controlled by CALL and RETURN instructions. One interrupt can be handled, but the second pushes the stack too far. The 8008 loses track of its programming, some random sequence of instructions is executed, and the CPU soon hits a HALT instruction.

The designer interested in a thorough discussion of interrupts and the 8008 should see Chapters 16 and 17 of Martin Research's book, *MICROCOMPUTER DESIGN.*

SEC. 6 CONTROL KEYS

The two Console board control keys, *M* and *=K,* are analogous to shift keys on a typewriter. Pressed at the same time as some other key on the Console keyboard, they can be used to create codes different from the basic interpretation of that key. A further explanation appears in the Model 420 Console board data sheet. Though not used in routine *MIKE 2* operations, the control keys can be programmed by the user as desired. These keys are used in the Monitor programs for other computers in the *modular micro* family.

SEC. 7 PROGRAMMING

A study of the program listing for the *MIKE 2* Monitor reveals that there are a number of useful subroutines there, which may be *called* by other programs as needed. A useful example is provided by the RAM test program (in the *Software* section of the *MIKE 2* Manual), which uses subroutines for displaying digits on the 420 Console board, and for incrementing the H and L registers.

Other useful software information will be added to that section of the Manual as it is supplemented from time to time.

The *401 CPU* is a central processing unit on a single printed circuit board. Based on the 8008 microprocessor, the 401 circuitry generates the essential control signals needed throughout the *MIKE 2* modular microcomputer system.

Features include a reliable crystal-controlled clock; a flexible three-state data bus, with bus drivers to allow full system expansion; full memory address capability; and specialized control signals which simplify system expansion by minimizing the extra interface circuitry needed on each new memory or input/output board.

SPECIFICATIONS

Speed

20 us instruction time, standard version.
12.5 us (60% faster), 8008-1 version.

Instructions

Full complement of 48 basic 8008 instructions.

Memory capacity

Supports full 8008 memory addressing capability:
16K bytes.

Interfacing

Connects directly to the modular micro bus with standard fifty-pin connectors. Three-state data bus drivers sink up to 40 MA; memory address registers (DH and DL) drive up to 10 TTL loads.

Power

Requires +5 V, 370 MA; -9 V, 40 MA; regulated ± 5%.
(-9 V can be derived from a -12 V supply using an optional on-board zener diode.)

OPTIONS

The *Short Stack* option boosts the interrupt-handling ability of the 8008 microprocessor. Plugging into an associated breadboard, four integrated circuits make up a two-byte stack, allowing the *MIKE 2* Monitor program to store registers and flags temporarily upon receipt of an interrupt. Since the *MIKE 2* Monitor also uses page 013 in RAM memory for interrupt handling, a version of the *Short Stack* option which includes the necessary RAM chips is available as well, at nominal cost. *(For details, see separate product literature on the Short Stack, KT9000.)*

9	8	7	6	5	4	3	2	Rev.	ECO	By	Ck	Date
A	A	A	A	A	A	A	A	A	016	BR	RHL	12.8.75

TECHNICAL DATA

SEC. 1 CIRCUIT DESCRIPTION

1.1 *Reference to Model 441 CPU Board* The Model 401 CPU board is substantially identical to the Model 441, which is a later version of the same circuitry. The 441 differs mainly in having space on the board for the Short Stack option (KT9000) and the 16-Channel Scope Display option (KT9007). In order to minimize duplication of information, only those details in which the 401 differs from the 441 are discussed here. The complete 401 schematic appears in the pages that follow. For a more detailed circuit description, the reader is referred to the 441 data sheet.

1.2 *Main Timing* The 8008 microprocessor requires a two-phase clock signal (ϕ1 and ϕ2). The 401 CPU board derives the 500 KHz signals required for the standard 8008 CPU from a 16.0 MHz crystal oscillator, composed of two 7404 inverters. One-half of a 7474 dual-D flip-flop functions as a divide-by-two counter, producing an 8.0 MHz square wave. This in turn is coupled to a 7493 divide-by-16 counter, which provides four outputs at 4.0, 2.0, 1.0, and 0.5 MHz. The last two are gated together to produce two 500 KHz signals, each with a 25% duty cycle, with the positive pulse of ϕ1 occurring midway between the positive pulses of ϕ2. A timing diagram appears in the 441 data sheet.

The 8008-1, a premium device, requires a clock with a repetition rate of 800 KHz rather than 500 KHz (60% faster). In order to modify the 401 CPU board for 8008-1 operation, it is necessary to (1) replace the 8008 microprocessor with an 8008-1; (2) replace the 16.0 MHz crystal with a 12.8 MHz crystal; (3) remove the 7474 flip-flop from location F2 on the 401 board; and (4) connect the 7404 crystal oscillator directly to the 7493 counter, as indicated by the dotted line in the *Main Timing* section of the 401 schematic. This last step is accomplished easily with a wire jumper connecting pins 9 and 11 at the site of the removed 7474, at F2 on the board.

1.3 *Bus Drivers* Like other MOS microprocessors, the 8008 cannot directly drive the assortment of memory chips and other devices that are found on the bus of a complete microcomputer. For this reason, the 401 CPU board includes bus drivers on the board. In this regard the 401 is similar to the 441 CPU board.

The 401 CPU board contains space for either of two parts of bus drivers. Either one type, or the other--*but not both pairs*--may be used.

At locations A2 and B2 is room for two 16-pin bus drivers. In early production of the 401 CPU board, these spaces were stuffed with 3216/8216 bus drivers, manufactured by Intel Corporation. The D Revision of the 401 board substitutes the 8T28, for reasons of economy and availability. A minor wiring change is involved, and is detailed on the 401 schematic diagram. (Compare parts 1 and 4 of the 401 schematic.)

At locations C2 and D2 is space for two 20-pin bidirectional bus drivers, type 74S241. Production of these Texas Instruments parts was delayed, and although these ICs would work

modular micros martin research

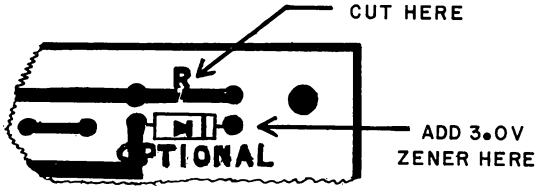
SEC. 1 CIRCUIT DESCRIPTION (cont'd)

in this application, production units of the 401 CPU board have always used the 16-pin devices instead.

1.4 *Short Stack Option* Formerly referred to as the *Swap Register,* the Short Stack option is needed to improve the interrupt-handling capabilities of the 8008 microprocessor. Details on circuit operation, with a schematic, are printed in a separate data sheet on the Short Stack option, KT9000.

Because there is no convenient place on the 401 CPU board for installation of the Short Stack, it is recommended that the 401 user who desires this feature install the four ICs on the breadboard that comes with the MIKE 2 system. Structurally, the option consists of a combined input/output port; no direct connection to the CPU board is necessary, and the circuit can be connected to the MIKE 2 bus at any available location in the system.

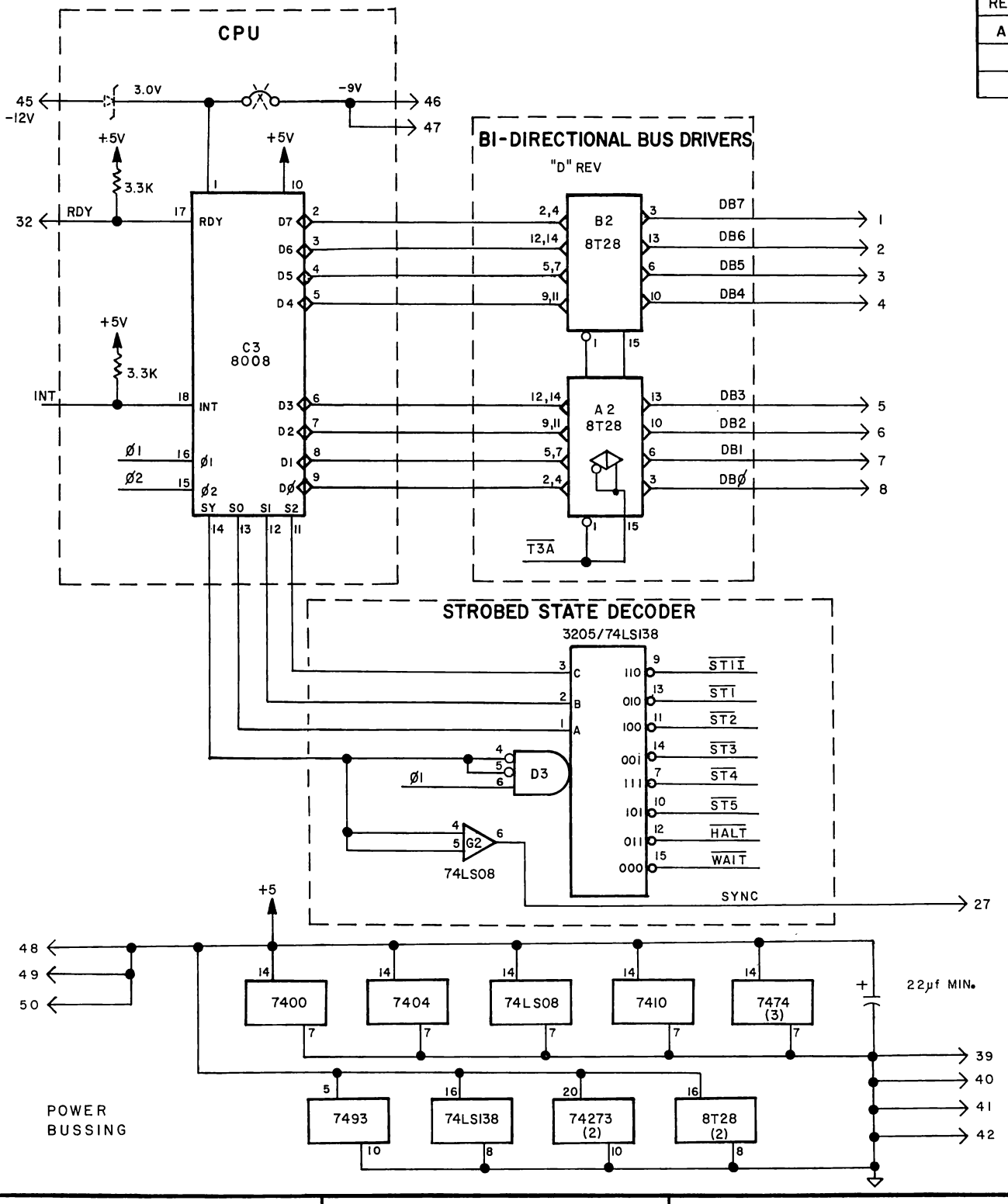
1.5 *-12 V Option* The 8008 microprocessor requires power supply voltages of +5 V and -9 V, while all other circuitry on the 401 board requires a single +5 V supply. Pins 46 and 47 on the *modular micro* bus may be used to provide the -9 V. (See the *CPU* section of the 401 schematic, Part 1.) However, if the user desires to power the 401 board from a -12 V supply, at pin 45 on the bus, this may easily be done.



(1) Holding the 401 board component side up, locate the upper right-hand corner area, as illustrated in the above drawing.

(2) Using a knife or other sharp edge, cut through the thin area of the foil wiring, immediately below the letter "R," as shown in the drawing. This disconnects the 8008 from the -9 V line on the bus.

(3) Install a 3.0 V zener diode, as shown in the drawing. Be sure to observe the proper polarity of the diode, with the band pointing to the right. Some diodes may show the diode symbol--an arrow--instead of a band, in which case the arrow should point to the right, as shown in the drawing. Or, the diode may have a "+" symbol; the diode should be inserted with the + on the right.

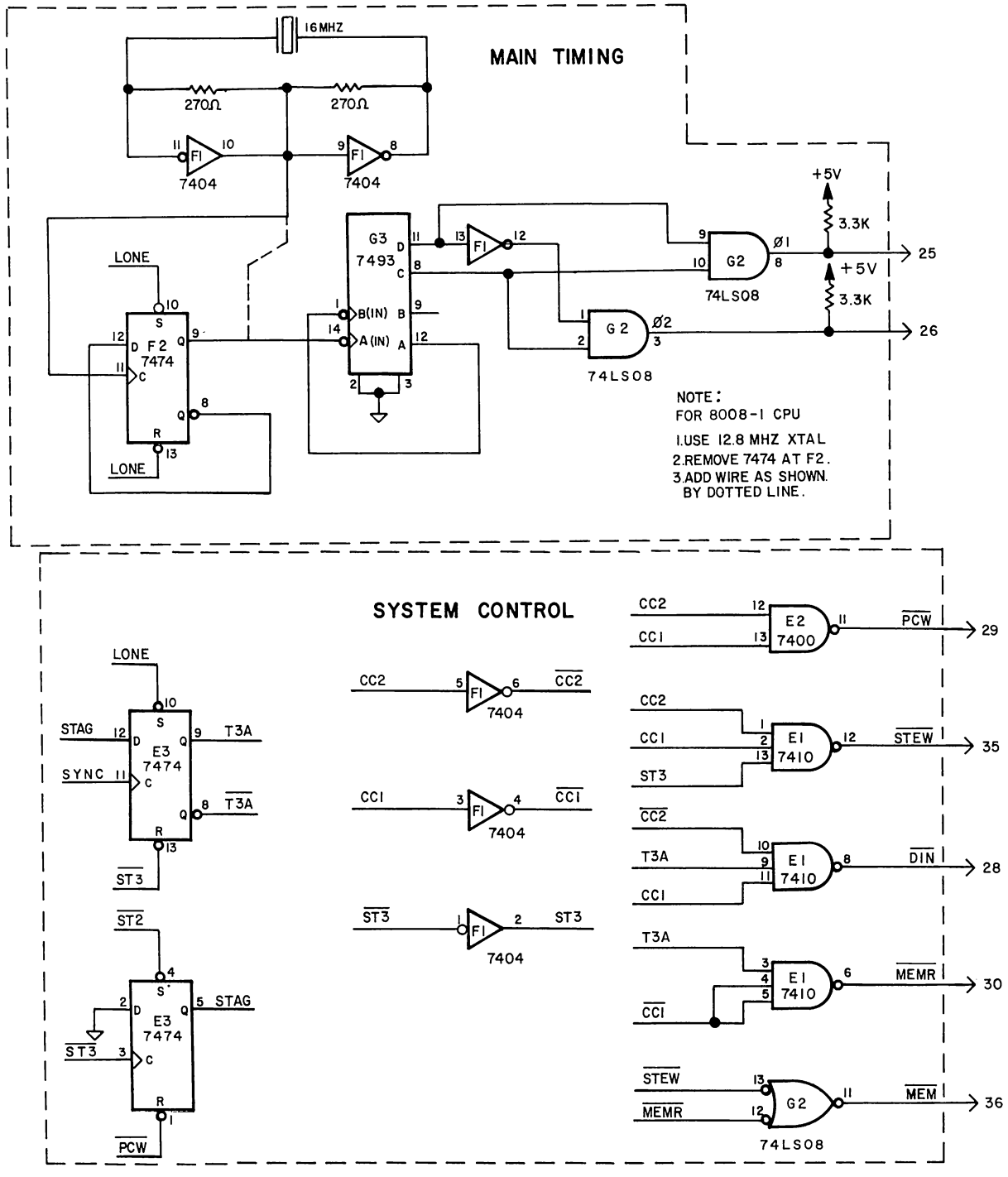


NOTE: PROPRIETARY INFORMATION. NOT TO BE DISCLOSED WITHOUT EXPRESS WRITTEN PERMISSION.	DRAWN BY Schneider	CHECKED BY DMK	DRAWING NO. DS 401 (Page 4)
	DATE 11-18-75	DATE 11-28-75	

EF4A

modular micros martin research

REV
A



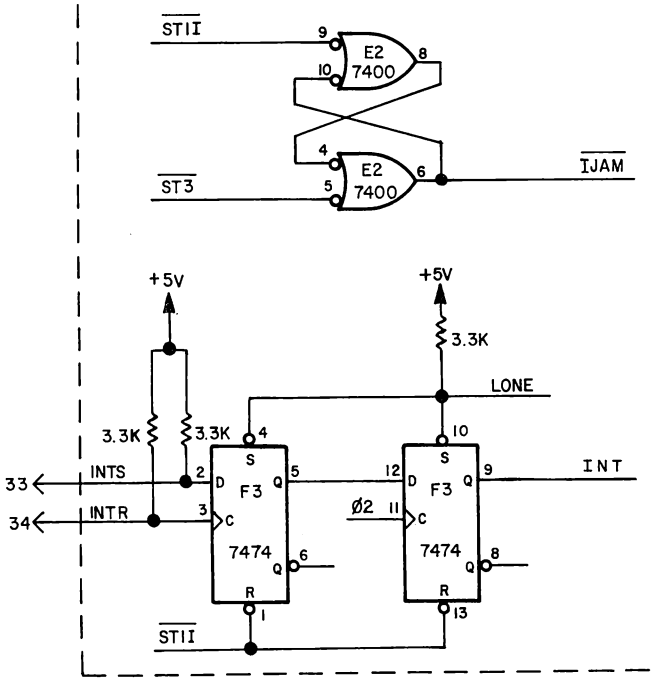
NOTE: PROPRIETARY INFORMATION. NOT TO BE DISCLOSED WITHOUT EXPRESS WRITTEN PERMISSION.	DRAWN BY Schneider	CHECKED BY DK	DRAWING NO. DS 401 (Page 5)
	DATE 11-18-75	DATE 11-28-75	

EF4A

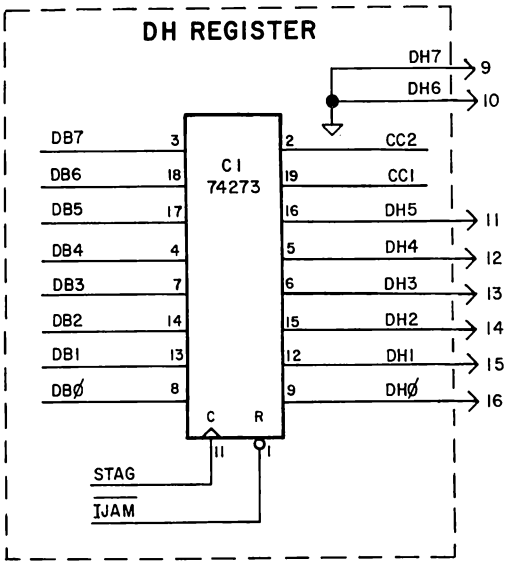
modular micros
martin research

REV
A

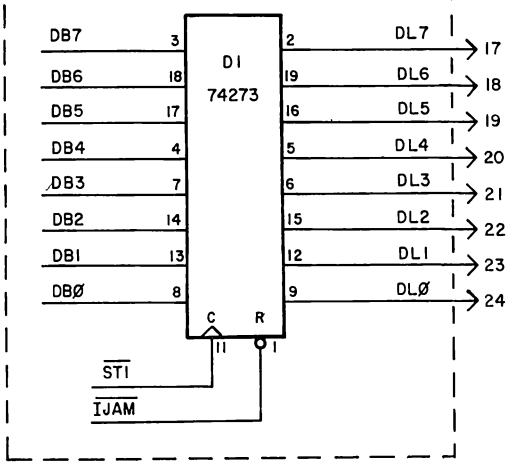
INTERRUPT LOGIC



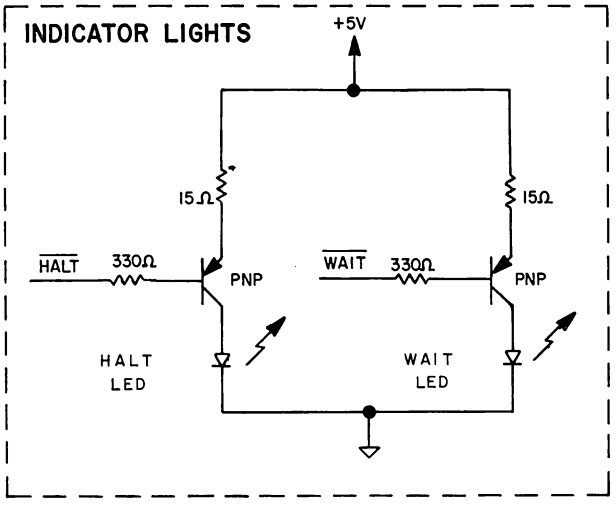
DH REGISTER



DL REGISTER



INDICATOR LIGHTS



NOTE: PROPRIETARY INFORMATION. NOT TO BE DISCLOSED WITHOUT EXPRESS WRITTEN PERMISSION.	DRAWN BY Schneider	CHECKED BY DHK	DRAWING NO. DS 401 (Page 6)
	DATE 11-17-75	DATE 11-28-75	

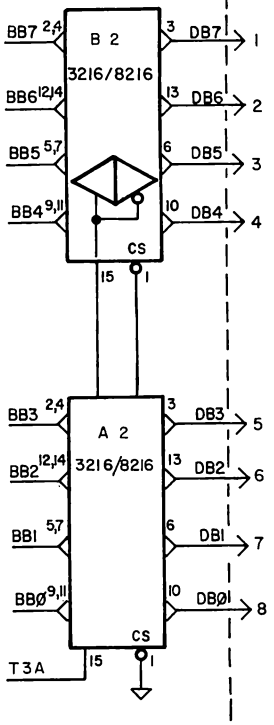
EF4A

modular micros martin research

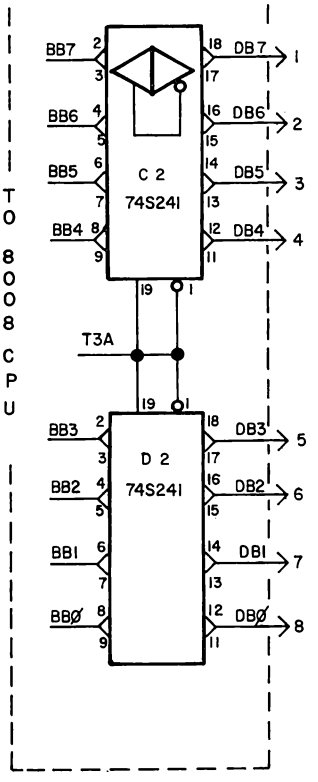
REV
A

BI-DIRECTIONAL BUS DRIVERS

"C" REV AND EARLIER



ALTERNATE BI-BUS DRIVERS



NOTE: PROPRIETARY INFORMATION. NOT TO BE DISCLOSED WITHOUT EXPRESS WRITTEN PERMISSION.	DRAWN BY Schneider	CHECKED BY DMK	DRAWING NO. DS 401 (Page 7)
	DATE 11-17-75	DATE 11-28-75	

EF4A

SEC. 2 PARTS LIST

2.1 *Standard Parts* The following parts are included with the 401-3 CPU board. This version of the 401 includes all necessary parts, including the 8008 microprocessor. Optional parts are shown in following sections.

Part No.	Quantity	Description
BD401	1	PC BOARD, central processing unit, for 8008 microprocessor
1000	1	CONNECTOR, PC board header, 50-pin, rt. angle, solder leads
1012	8	SOCKET, 14-pin DIP, solder-tail, tin
1013	2	SOCKET, 16-pin DIP, solder-tail, tin
1014	1	SOCKET, 18-pin DIP, solder-tail, tin
1015	2	SOCKET, 20-pin DIP, solder-tail, tin
2000	1	IC, 7400 quad NAND gate
2004	1	IC, 7404 hex inverter
2007	1	IC, 74LS08 quad AND gate
2008	1	IC, 7410 triple 3-input NAND
2029	3	IC, 7474 dual D flip-flop
2035	1	IC, 7493 4-bit binary counter
2042	1	IC, 74LS138 3-to-8 decoder
2083	2	IC, 74273 octal latch
2130	2	IC, 8T28B quad bi-bus driver (early version used P/N 2133, 8216: see Sec. 1.3 above)
2202	1	IC, 8008 central processing unit
3000	2	TRANSISTOR, 2N5138 PNP
3500	1	LED, red, with red frosted lens
3501	1	LED, red, with clear lens
4000-27	2	RESISTOR, 15 ohms, 10%, ¼w (color code brown, green, black)
4000-42	2	RESISTOR, 270 ohms, 10%, ¼w (color code red, violet, brown)
4000-43	2	RESISTOR, 330 ohms, 10%, ¼w (color code orange, orange, brown)
4000-55	7	RESISTOR, 3.3K, 10%, ¼w (color code orange, orange, red)
5502	1	CAPACITOR, 22 to 50 uf, 15 V min., tantalum
5900-2	1	CRYSTAL, 16.0 MHz

2.2 *8008-1 Option* The following parts are used with the 401-5, a version of the CPU board based on the premium 8008-1 microprocessor, which features a speed 60% faster than the standard 8008. (For instructions see Sec. 1.2.)

Part No.	Quantity	Description
2203	1	IC, 8008-1, premium CPU (replaces P/N 2202, 8008)
5900-1	1	CRYSTAL, 12.8 MHz (replaces P/N 5900-2, 16.0 MHz)
-----	1	JUMPER: any hookup wire, of suitable length, may be used (replaces 1 P/N 2029, 7474)

SEC. 2 PARTS LIST (cont'd)

2.3 *-12 V Power Option* The following part may be used to provide the 401 CPU board's -9 V power requirement from a -12 V supply. (See Sec. 1.5 above.)

Part No.	Quantity	Description
3700-05	1	DIODE, IN5225 zener, 3.0 V, 5%, 500 mw

The *441 CPU* is a central processing unit on a single printed circuit board. Based on the 8008 microprocessor, the 441 circuitry generates the essential control signals needed throughout the *MIKE 2* modular microcomputer system.

The 441 supports an exclusive *16-Channel Scope Display* option, which draws timing diagrams of all important *MIKE 2* system signals (up to 16) on an external triggered-sweep oscilloscope. Useful in debugging and for educational purposes, this inexpensive option compares in usefulness to digital display consoles costing thousands of dollars.

The *Short Stack* option, another valuable feature, boosts the interrupt-handling ability of the 8008 microprocessor. Plugging into designated spaces on the 441 CPU board, four integrated circuits make up a two-byte stack, allowing the *MIKE 2* Monitor program to store registers and flags temporarily upon receipt of an interrupt. Since the *MIKE 2* Monitor also uses page 013 in RAM memory for interrupt handling, a version of the *Short Stack* option which includes the necessary RAM chips is available as well, at nominal cost. *(For details, see separate product literature on the Short Stack, KT9000.)*

A reliable crystal-controlled clock is standard with the 441; its speed may be set for either the 8008 or the 8008-1 (60% faster) by interchanging one inexpensive part, but without changing the crystal. Other standard features: a flexible three-state data bus, with bus drivers to allow full system expansion; full memory address capability; and specialized control signals which simplify system expansion by minimizing the extra interface circuitry needed on each new memory or input/output board.

SPECIFICATIONS

Speed	20μs instruction time, standard version. 12.5μs (60% faster), 8008-1 version.
Instructions	Full complement of 48 basic 8008 instructions.
Memory capacity	Supports full 8008 memory addressing capability: 16K bytes.
Interfacing	Connects directly to the modular micro bus with standard fifty-pin connectors. Three-state data bus drivers sink up to 40 MA; memory address registers (DH and DL) drive up to 10 TTL loads.
Power	Requires +5 V, 370 MA; -9 V, 40 MA; regulated ±5%. (-9 V can be derived from a -12 V supply using an optional on-board zener diode.)

TECHNICAL DATA

SEC. 1 CIRCUIT DESCRIPTION

1.1 *Introduction* The 441 CPU board, based on the 8008 microprocessor, is normally the most complex element in the microcomputer. The central processor controls nearly all of the signals which appear on the system's bus structure.

1.2 *The 8008 CPU* The 8008 microprocessor sends and receives data in eight-bit parallel format via its data bus terminals, labeled *D7* through *DO* in the 441 schematic (pages 4 and 5). These terminals are used to:

(1) *Read instructions* from the system's program memory, either from ROM (permanently-stored *read-only memory*), or from RAM (*random access memory*);

(2) *Write* new information into memory, usually RAM;

(3) *Read* information from the system's memory, where it has previously been stored;

(4) *Receive* information from digital logic sources outside the CPU (i.e., from *input ports*);

(5) *Send* information to destinations outside the CPU (i.e., to *output ports*); and

(6) *Address* a desired word in memory--from which information is to be read, or into which data is to be written, or from which a new instruction is to be fetched.

The two high-order data bus terminals, D7 and D6, are also used to send out signals which control memory read/write and input/output cycles.

These multiple functions imply a complex time-sharing of the eight-bit data bus. A great deal of the necessary timing logic is included within the 8008 microprocessor itself. For details on internal processor operations, the reader is referred to the 8008/8008-1 data sheet, DS2202/2203.

In addition, a working 8008 microcomputer requires a certain amount of digital control circuitry external to the microprocessor itself. This is principally because the 8008 package allows only a relatively small number of connections to the outside world--18 terminals total, including power--which therefore must handle data and timing signals in a highly encoded form.

©1975, Martin Research. Pp. 1-16, Rev. A, Nov. 1975.

REV. A

SEC. 1 CIRCUIT DESCRIPTION (cont'd)

To begin with, the CPU board must supply the 8008 with a two-phase *clock,* ϕ1 and ϕ2. In turn, the 8008 generates four special timing signals--*SY, SO, S1,* and *S2*--which are decoded externally to produce a number of specialized signals, synchronizing memory and input/output ports with the microprocessor.

Additional external circuitry is necessary for the 8008 to interact with the system's memory. A location in memory which is to be written into, or read from, must be provided simultaneously with its correct address, *and* with a data path to the microprocessor. Obviously the eight-bit 8008 data bus cannot perform both functions at the same instant. Therefore, the CPU board includes *address registers* external to the 8008. In a typical cycle, the 8008 data bus first sends out eight bits of memory address, which are stored in the *DL register.* Second, the 8008 data bus transmits six more address bits, which are latched up in the *DH register.* The fourteen address lines, DH5 through DHO and DL7 through DLO, form an *address bus,* which--like the data bus itself--is piped throughout the system via the fifty-wire flexible cable. When the third stage of the cycle begins, the desired memory location has already been selected by the address bus, and the information flows between memory and microprocessor via the data bus.

Input and output ports impose a similar timing requirement. The I/O information being transferred cannot appear on the 8008 data bus at the same time as the address of the I/O port being selected. In this case, the I/O port address appears on the DH address bus. Information flows from *input* ports into the 8008 via the data bus, as would be expected. However, information flowing outward from the 8008 to an *outport* port travels via the DL bus.

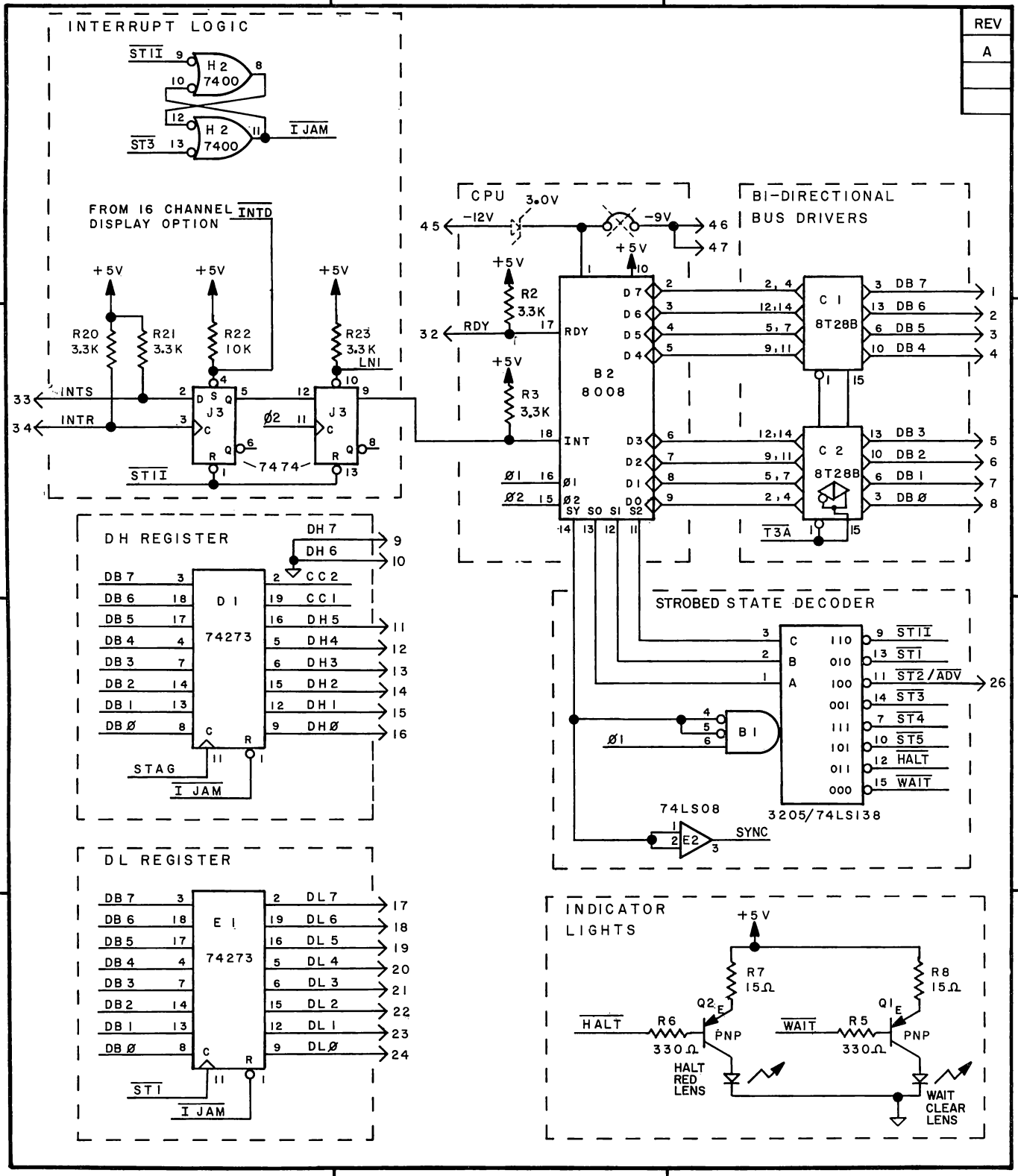
Because the 8008's ultraminiaturized internal circuitry is not designed to drive the large array of memory and other devices typically found on the microcomputer's data bus, the 441 CPU board includes *bus drivers* to amplify the output power of the 8008.

And, additional external circuitry supports the system *interrupt* function, necessary in starting up the 8008, and useful in ensuring that a high-priority task gets immediate attention.

The above outline demonstrates the need for circuitry external to the 8008 on the CPU board. The remainder of this circuit description focuses on the hardware design of the 441 board, section by section, following the circuit breakdown indicated by the dotted blocks in the schematic diagram.

1.3 *Main Timing* The 8008 microprocessor requires a two-phase clock operating at 500 KHz, while for the premium 8008-1, the clock runs at 800 KHz, 60% faster.

The 441 CPU board employs a crystal oscillator as its basic timing element. (See the *Main Timing* section of the 441 schematic.) Two 7404 TTL inverter stages are coupled together in an unstable configuration. The crystal, resonant at 16.0 MHz, determines the frequency of oscillation.

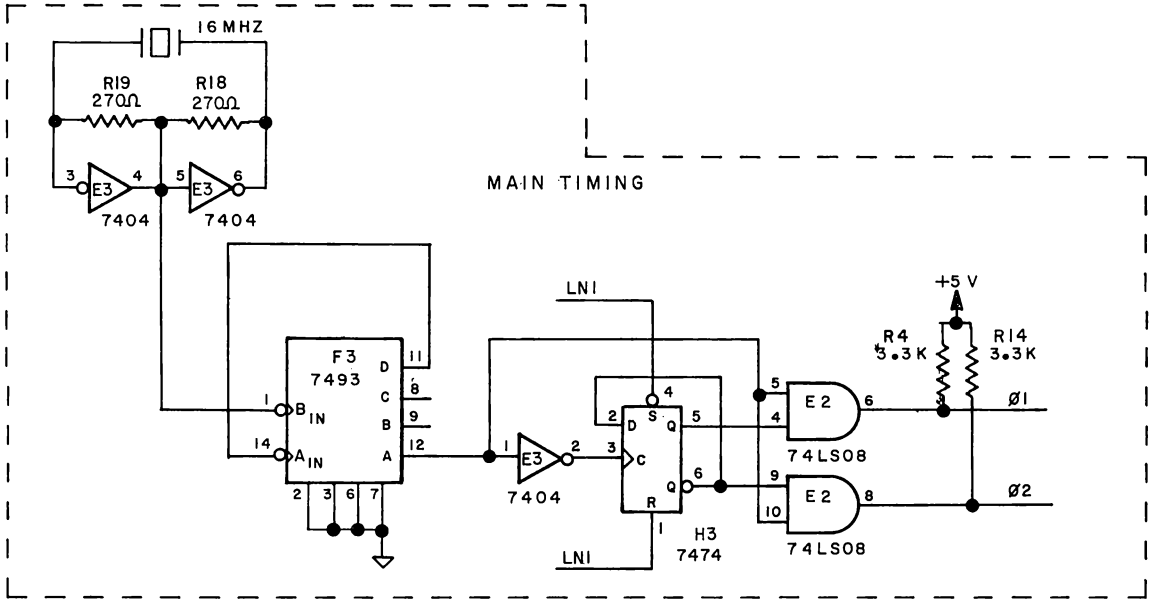


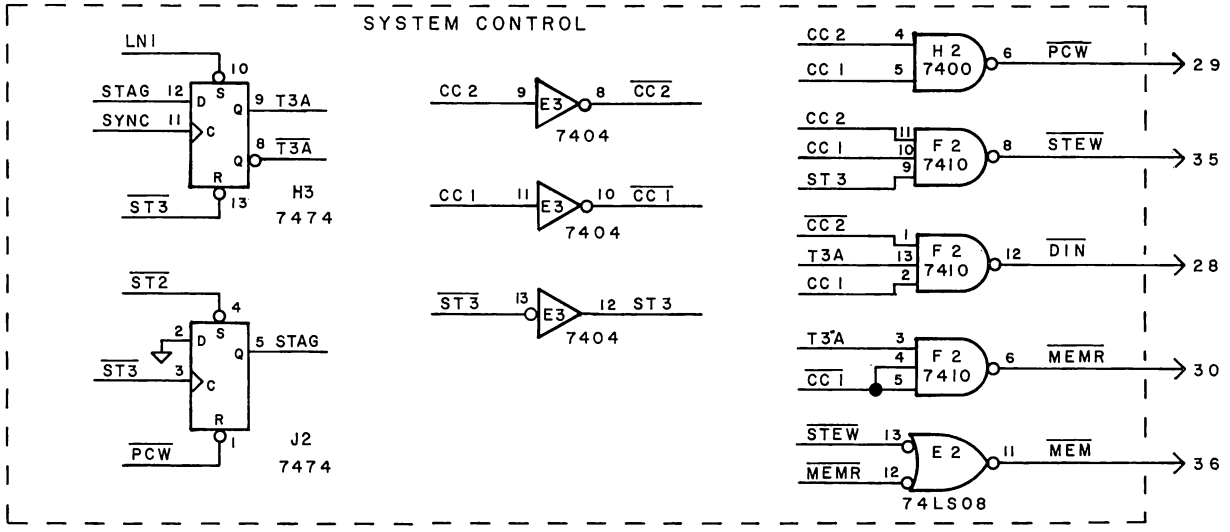
NOTE: PROPRIETARY INFORMATION. NOT TO BE DISCLOSED WITHOUT EXPRESS WRITTEN PERMISSION.	DRAWN BY ROECK DATE 11-17-75	CHECKED BY DNA DATE 11/17/75
		DRAWING NO. DS 441 (Page 4)

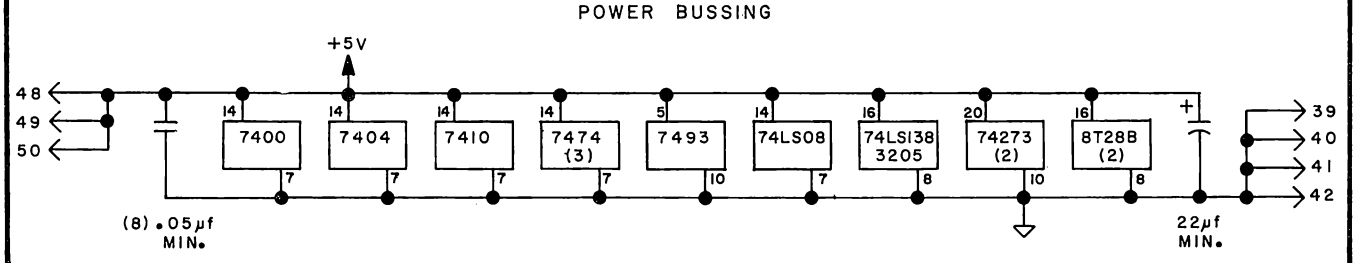
EF4A

modular micros
martin research

REV
A







1	2	3
NOTE: PROPRIETARY INFORMATION. NOT TO BE DISCLOSED WITHOUT EXPRESS WRITTEN PERMISSION.	DRAWN BY ROECK DATE 11-17-75	CHECKED BY DK DATE 11/17/75	DRAWING NO. DS 441 (Page 5)

EF4A

modular micros martin research

REV. A

SEC. 1 CIRCUIT DESCRIPTION (cont'd)

The 16.0 MHz signal is connected to the *B* input of a 7493 four-bit binary counter. The *B, C,* and *D* stages successively divide their input frequencies by two, producing signals at the *B, C,* and *D* outputs of 8, 4, and 2 MHz respectively. The 2 MHz signal is coupled to the *A* input of the same chip, where it is again divided by two, producing a 1 MHz signal at the *A* output.

A 7474 flip-flop serves as a final divide-by-two counter. This D-type device moves the state at its *D* input into its *Q* output as its clock (*C*) goes from 0 to 1. Here, the inverted *Q* output (inversion is indicated with a bubble) is connected to the *D* input, causing the *Q* output to change to its opposite logical state each time the device is clocked. This action effectively divides the input frequency by two, producing a 0.5 MHz (500 KHz) output.

The unused 7474 set (*S*) and reset (*R*) pins are held at logic high level by the *LN1* (logic one) bus. This line is connected to +5 V through a resistor, protecting the TTL inputs against supply transients.

In order to keep the 7474 (a rising-edge-triggered device) in phase with the outputs from the 7493 (whose outputs change on the falling edge of the clock), an inverter precedes the 7474.

The 500 KHz and 1.0 MHz signals serve as inputs to a 74LS08 AND gate, whose output is high only when both inputs are high. Another AND gate combines the 1.0 MHz signal and the *inverse* of the 500 KHz signal. Fig. 1.3.1 illustrates how this logic produces a symmetrical two-phase clock, $\phi 1$ (phase one) and $\phi 2$ (phase two), appearing as positive pulses with alternating phases.

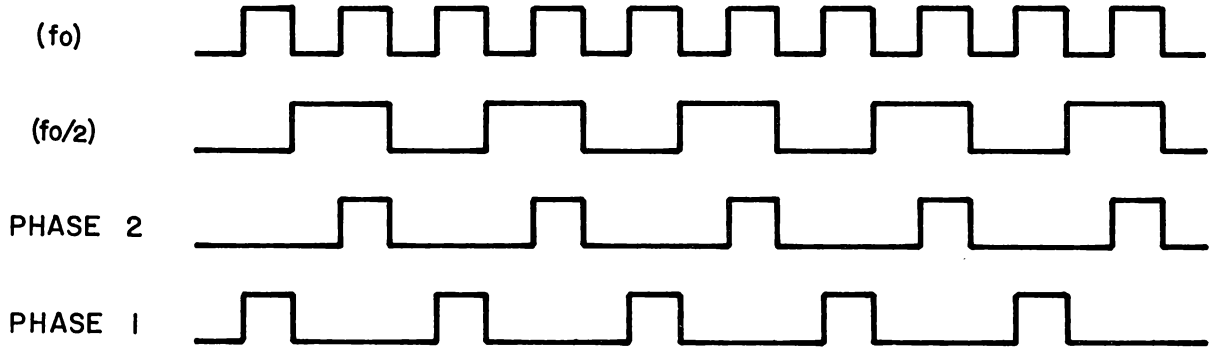


Figure 1.3.1--Two-Phase Clock Generator Timing Diagram

3.3K pullup resistors at the 74LS08 outputs ensure that their logic high levels meet the 3.5 V requirements of the 8008's MOS circuitry.

REV. A

SEC. 1 CIRCUIT DESCRIPTION (cont'd)

1.4 *8008-1 Clock Conversion* In order to convert the 441 CPU board for 8008-1 operation, it is necessary to provide an 800 KHz clock. This is easily done by removing the 7493 divide-by-16 counter at location F3 on the board, and replacing it with a *suitable* 7490 divide-by-10 counter. The 7490 is a pin-compatible counter whose *B-C-D* section repeats after every fifth count, rather than every eighth count. Thus the 7490 *D* output is at 16/5 or 3.2 MHz, and its *A* output is at 1.6 MHz. This frequency, which compares with 1.0 MHz when the standard 7493 is used, produces ϕ1 and ϕ2 signals at the required 800 KHz.

Warning: a survey of manufacturers' data sheets reveals that many 7490 counters are rated for *B* clock speeds of less than 16.0 MHz; and, in practice, some fail to count in this circuit. The Texas Instruments SN7490A is appropriately rated and works in this application. To avoid disappointment, this specification should be checked before purchase. A part specifically tested for this circuit is available from Martin Research (see the Parts List, Sec. 2 below).

1.5 *Strobed State Decoder* The 8008 divides the ϕ2 signal by two, internally, and produces a signal called *SY.* A 74LS08 gate serves as a buffer for increased driving power; the amplified signal is labeled *SYNC.* The 8008 also creates three *state signals* called *S2, S1,* and *S0.* These signals indicate which of eight possible *states* the 8008 is presently in--including T1, T2, T3, T4, and T5. (The remaining three states--*T1I, HALT,* and *WAIT*--are discussed later). In the course of normal operations, the 8008 goes through the states T1 through T5, repeating continually--and sometimes leaving off T4 or T5 when these states are unnecessary. Referring to the 441 schematic, the *Strobed State Decoder* stage consists of a 3205/74LS138 3-to-8 decoder which decodes the three state signals. With its eight outputs normally at logic high levels, only one can be low at once--as selected by the 8008 state signals at the 'LS138's *C, B,* and *A* inputs. An additional condition necessary to produce an active-low output signal is that the chip enable terminals be in the proper states. Since SY is connected to the two active-low (bubbled) enable terminals, and ϕ1 to the active-high terminal, a selected decoder output can go low only when ϕ1 is high, and SY low.

Thus, as the 8008 cycles through the T1 through T5 states, the 'LS138 produces a series of five related *strobe* signals, labeled $\overline{ST1}$ through $\overline{ST5}$ (the bar above the signal indicating that it is active low). These signals are synchronized with ϕ1 of the 8008 clock, and with the low level of the SY signal.

For a timing diagram, see Fig. 1.5.1. Note that $\overline{ST1}$ occurs during T1 time, $\overline{ST2}$ during T2 time, and so forth. For comparison, note the three-digit binary numbers labeling the outputs *within* the outline of the Strobed State Decoder, in the 441 schematic. For example, the binary digits *010* label the strobe $\overline{ST1}$. This indicates that this strobe occurs when S2, S1, and SO are 0, 1, and 0 respectively--a fact confirmed by inspection of the timing diagram, Fig. 1.5.1.

REV. A

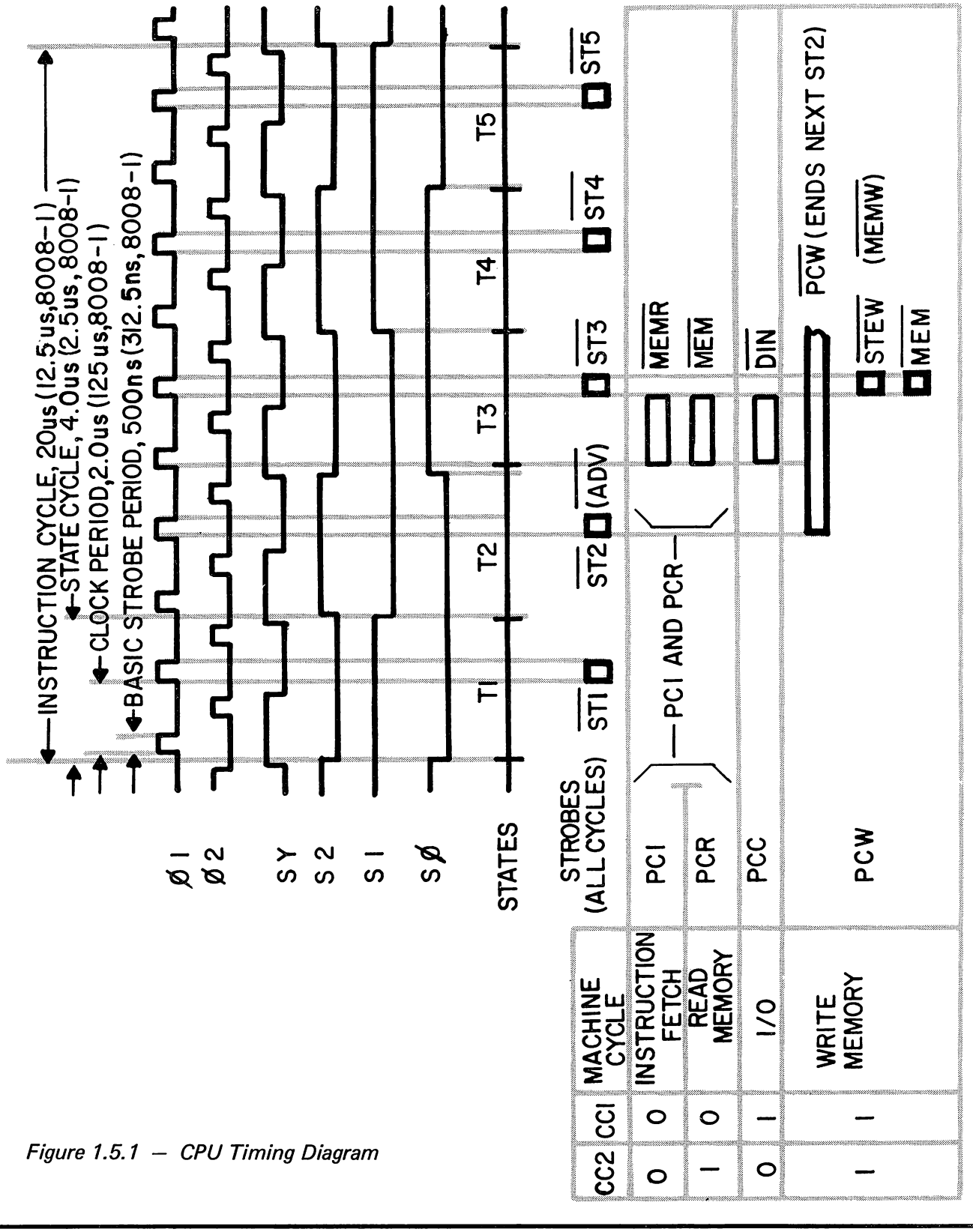


Figure 1.5.1 — CPU Timing Diagram

SEC. 1 CIRCUIT DESCRIPTION (cont'd)

The locations of these strobes within their respective states have been chosen for proper synchronization of the 8008 with external circuitry. As with any bus-structured system, there are appreciable delays between the beginning of a timing state, and the time when new information on the data bus can be presumed to have settled down.

As stated earlier in general terms, the 8008 successively outputs different information on its data bus during different phases of its operations. To be specific: at T1 time, the 8008 outputs a low-order memory address, giving the location in memory which may later be read from or written into. At T2 time, the high-order six bits of the memory address come out on the data bus. So also do two control bits, called CC2 and CC1. Finally, during T3 time many of the important data transfers take place--between CPU and memory, and between CPU and input ports.

The T4 and T5 states are used for data transfers internal to the 8008, and are not generally of importance in the hardware design of the CPU board. Their use is illustrated in the chart on Internal Processor Operations in the 8008 data sheet.

1.6 *DH and DL Registers* In order to provide temporary storage for the memory addresses and control bits which appear on the 8008 data bus at T1 and T2 times, two eight-bit registers are supplied on the 441 CPU board. Called the *DH* and *DL registers,* each consists of a single 74273 octal latch.

The eight inputs of each register are connected to the 8008 data bus. In order to prevent overloading of the 8008, the buffered data bus, DB7 through DB0, is used. (The Bus Driver stage is discussed in Sec. 1.8.) The 74273 is loaded when its clock (*C*) terminal is brought from a 0 to a 1. The DL register is clocked with the $\overline{ST1}$ strobe, loading it at the end of the ST1 portion of T1 time. During T2 time, the DH register is loaded by the signal called *STAG.* (STAG goes high at the beginning of the $\overline{ST2}$ strobe: see Sec. 1.7.) Thus, by the middle of T2 time, the DH and DL registers contain a memory address which may be used later in the cycle for reading from or writing into memory. This fact is recognized by an alternate name for the $\overline{ST2}$ strobe, namely, $\overline{ADV}$, for *address valid.*

Note in passing that the DH and DL registers may be reset to zero via the reset (*R*) terminals. The $\overline{IJAM}$ signal which does this occurs during interrupt handling (Sec. 1.10 below).

The sixteen DH and DL register outputs--DH7 through DH0, and DL 7 through DL0--constitute an *address bus* which appears on terminals 9 through 24 of the 50-pin bus piped throughout the *modular micro* system. Note in the schematic that the *DH7* and *DH6* terminals are simply connected to ground. Effectively only the fourteen bits from DH5 through DH0, and DL7 through DL0, are used to address memory. This allows the 8008 microcomputer to use 2^{14} or 16 K words of memory, more than most practical micro designs would ever call for.

Since the two high-order bits on the 8008 data bus at T2 time are really *control bits, CC2* and *CC1,* properly speaking the "DH" register should be called the *CC-DH register.* In any case,

REV. A

SEC. 1 CIRCUIT DESCRIPTION (cont'd)

CC2 and CC1 indicate which of four possible types of *machine cycle* the 8008 is currently in. These machine cycles are shown in Fig. 1.5.1, and consist of the *PCI* (instruction fetch) cycle; the *PCR* (read memory) cycle; the *PCC* (control, or I/O cycle); and the *PCW* (memory write) cycle.

In the course of executing a complete instruction, the 8008 always goes through a PCI cycle first, fetching from memory the eight-bit word which contains the instruction in encoded form. At PCI-T1 time, the low-order memory address, of the location in memory where the instruction is to be fetched, is latched up in the DL register. During PCI-T2 time, the high-order memory address of the instruction location is latched up in the CC-DH register. At the same time, the control bits (CC2, CC1) appear, and since this is a PCI cycle, they will read *00*. During PCI-T3 time, the circuitry generates a memory read pulse (discussed later) which causes the instruction code to be transmitted to the 8008. The 8008 decodes this instruction to find out what operation is called for. In most cases, another machine cycle follows--either a PCC (in the case of I/O instructions), or a PCR (when memory is to be read), or a PCW (when memory is to be written into). Some instructions call for three machine cycles.

Note that the instruction is well underway before the 8008 "knows" what the instruction is--only when the control bits, CC2 and CC1, are latched up at PCI-T2 time.

1.7 *System Control Circuitry* Once the 8008 has decoded its instruction, it must usually carry out data transfers with memory or an I/O port. To do this, the 441 CPU board must generate a variety of special purpose strobe and enable signals. The circuitry which does this, shown in the *System Control* section of the 441 schematic, consists of two flip-flops and a collection of combinational logic gates.

These signals are associated with *T3A* time in the 8008 machine cycle, when many important data transfers take place. The timing diagram in Fig. 1.7.1 illustrates the relation of T3A to the SY, $\overline{ST2}$, $\overline{ST3}$, and $\overline{ST4}$ signals which have been discussed earlier.

STAG is a building-block signal used for generating the *T3A* family of controls. $\overline{ST2}$ sets a 7474 flip-flop, and the rising edge of $\overline{ST3}$ clocks the flip-flop while the D input is held at ground, resetting the flip-flop. Thus, STAG (shown on line 5, Fig. 1.7.1) is normally a positive pulse starting at the beginning of ST2 and finishing at the end of ST3.

Another 7474 uses the STAG signal to create T3A. Every time that SYNC goes high, it clocks this flip-flop. If STAG, at the D input, is high, T3A goes high at the output. A short time later, $\overline{ST3}$ comes along and resets T3A to zero. Thus, T3A is a positive pulse which begins near the beginning of T3 time, and ends when ST3 begins. (See line 6 in Fig. 1.7.1.)

However, this is the situation *only* during PCI, PCR and PCC cycles. T3A time is that time when the 8008 data bus is open to receive data from the outside world--from memory or from an input port. During PCW cycles, when the CPU needs to drive external memory, no T3A pulse should be created. Consider therefore the creation of the $\overline{PCW}$ signal (line 7, Fig. 1.7.1).

REV. A

SEC. 1 CIRCUIT DESCRIPTION (cont'd)

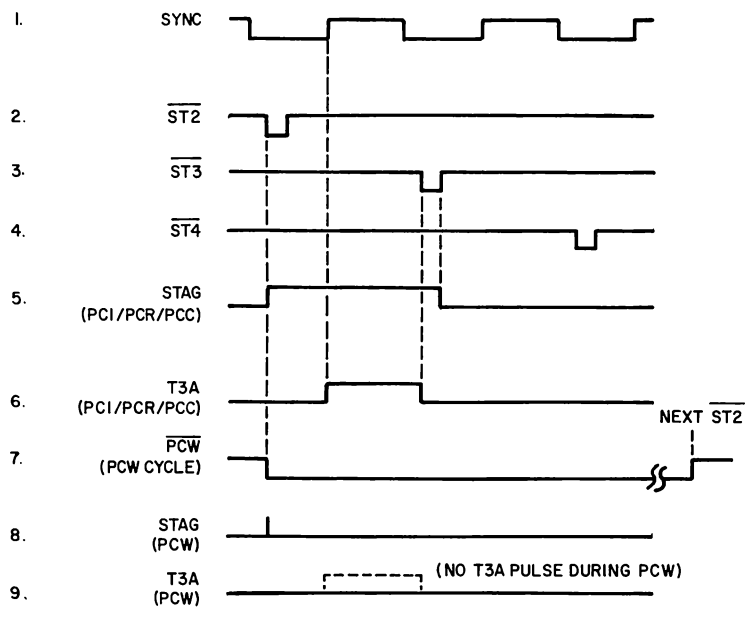


Figure 1.7.1 — T3A Timing Diagram

During PCW (memory write) cycles, the STAG flip-flop is set as usual by $\overline{ST2}$. The resultant STAG pulse clocks the 74273 which makes up the CC/DH register, and, since this is a PCW cycle, the CC2 and CC1 outputs become *11.* This condition is detected by a 7400 NAND gate (shown at the top right-hand corner of the *System Control* section of the 441 schematic). This gate generates $\overline{PCW}$, an active-low signal indicating the presence of a PCW cycle. ($\overline{PCW}$ stays low until the next STAG pulse changes the CC2 and CC1 bits, at the next ST2 time.)

The $\overline{PCW}$ signal is also fed back to the reset (*R*) terminal of the STAG flip-flop, resetting STAG to zero. As a result, during PCW cycles the STAG pulse is just a short pulse, or *glitch.* By the time that SYNC goes high again, STAG has gone away. Thus the T3A flip-flop fails to generate a T3A pulse during PCW cycles (line 9, Fig. 1.7.1).

During both PCI (instruction fetch) and PCR (memory read) cycles, the CPU reads from memory at T3A time. Since $\overline{CC1}$ is low only during PCI and PCR, CC1 is combined with T3A to produce a memory read signal called $\overline{MEMR}$. This signal is used by memory boards to activate memory chips.

During PCC cycles, the CPU receives data from input ports during T3A time. The control bit code for PCC--*01*--is therefore combined with T3A by a 7410 three-input NAND gate to produce

modular micros martin research

SEC. 1 CIRCUIT DESCRIPTION (cont'd)

$\overline{DIN}$, a universal input/output strobe/enable signal. (Details on I/O interfacing are presented in the data sheet for KT9001, a universal I/O kit.)

During PCW cycles, the CPU writes into RAM memory during T3 time. Instead of using T3A time, the control circuitry combines the control code for PCW--*11*--with ST3, creating a memory write signal called *$\overline{STEW}$.* Derived originally from the name $\overline{ST3W}$, this signal is also sometimes labeled *$\overline{MEMW}$.*

Finally, the two active-low memory enabling signals, $\overline{STEW}$ and $\overline{MEMR}$, are read together to produce a common memory signal, *$\overline{MEM}$.* When this signal is low, it means that memory is either to be read *or* written into; it is useful in memory decoding circuitry.

Three 7404 inverter stages in the System Control circuitry assist in developing signals of proper polarity for use in the combinational logic.

1.8 *Bi-Directional Bus Drivers* Two 8T28 bus drivers are used in the 441 CPU board to help the 8008's MOS circuitry drive the data bus of a fully-configured microcomputer. These ICs have tri-state outputs, which may take on any one of three different levels: the normal TTL *high* state, near +5 V, and capable of *sourcing* relatively small currents; the normal TTL *low* state, near ground, capable of *sinking* somewhat larger currents; and a third *high-impedance* state, where the outputs draw extremely small currents, allowing the output voltage level to be determined by some other circuitry on the same bus.

In the 441, the 8T28s are connected so that when a given output is in the third state, it may be used as an input. The device now drives in the opposite direction. During most phases of 8008 operations, the 8T28s drive *outwards,* away from the 8008 and towards the outside world. However, during T3A time, $\overline{T3A}$ goes low, and the 8T28s drive inwards toward the 8008's internal circuitry.

1.9 *HALT and WAIT Indicator Lights* The 8008 may be made to cease executing a program with a HALT instruction. This causes the 8008 to cycle internally in the HALT state until the CPU receives an interrupt. The HALT condition is indicated by a special combination of the state signals, S2, S1, and S0, and causes the 441's state decoder to produce a string of pulses on the $\overline{HALT}$ strobe line. A PNP transistor switch lights up the red HALT light on the 441 CPU board. In order to end the HALT state, the user must cause the microcomputer to be interrupted.

The WAIT light goes on when the 8008's *RDY* (ready) line is brought low. Normally held to +5 V by a pullup resistor, this line may be forced low by an external memory board or I/O device whose speed is too slow for the CPU. The 8008 cycles internally in a special T3 WAIT state until the RDY line goes high again. An application example is provided by memory boards in the *modular micro* series which feature optional WAIT circuitry. However, memory components furnished by Martin Research are generally fast enough to avoid having to use this feature with the 441 CPU board.

REV. A

SEC. 1 CIRCUIT DESCRIPTION (cont'd)

1.10 *Interrupt Logic* When power is first applied to the microcomputer, the 8008 must receive an initial *interrupt* signal by bringing its *INT* terminal high. The CPU must also be interrupted to escape a HALT condition. And this feature is also used to interrupt a program currently being executed in order to jam in a higher-priority task.

Two 7474 flip-flops make up a pair of interrupt request latches. The first has two input lines--*INTS* and *INTR,* both of which appear on the *modular micro* bus and are available to circuitry throughout the system. Both lines are held high by pullup resistors to +5 V. Assuming the INTS line is high, the INTR line may first be pulled low, then released, clocking the flip-flop. The output goes high (pin 5), and at the next $\phi 2$ pulse, the second flip-flop goes high also. Synchronization with $\phi 2$ is required to avoid an interrupt race within the microprocessor, as detailed in the 8008 data sheet.

Pulling and holding the INTS line low prevents the INTR line from interrupting the microcomputer. A third line, $\overline{INTD}$, which also causes an interrupt, derives from the 16-Channel Scope Display option (described in a separate data sheet on the KT9007).

When the 8008 recognizes an interrupt, it substitutes a *T1I* state for the next T1 state. The state decoder produces an associated strobe, $\overline{ST1I}$, which serves to reset both 7474 interrupt request flip-flops. It also sets a set-reset flip-flop made up of two 7400 NAND gates (shown in the schematic as negative OR gates). The output is a signal called $\overline{IJAM}$, a negative pulse which clears the DH and DL registers to zero. The control bits are also zero--corresponding to a PCI (instruction fetch) cycle--and the 8008 now executes whatever instruction is stored at memory location zero. This would typically be a restart instruction, such as RST 020, which causes the CPU to jump to location 000 020, the beginning of the user's interrupt subroutine. An example of such a program is in the Monitor program supplied with the *MIKE 2* microcomputer.

In order to prevent an interrupt from disturbing registers being used by the main line program, the 441 supports a *Short Stack* option. The circuitry is described in a separate data sheet on the Short Stack, KT9000.

SEC. 2 PARTS LIST

2.1 *Standard Parts* The following parts are included with the 441-3 CPU board. This version of the 441 includes all necessary parts, including the 8008.

Part No.	Quantity	Description
BD441	1	PC BOARD, central processing unit, for 8008 microprocessor
1000	1	CONNECTOR, PC board header, 50-pin, rt. angle, solder leads
1012	8	SOCKET, 14-pin DIP, solder-tail, tin
1013	3	SOCKET, 16-pin DIP, solder-tail, tin
1014	1	SOCKET, 18-pin DIP, solder-tail, tin
1015	2	SOCKET, 20-pin DIP, solder-tail, tin

REV. A

SEC. 2 PARTS LIST (cont'd)

Part No.	Quantity	Description
2000	1	IC, 7400 quad NAND gate
2004	1	IC, 7404 hex inverter
2007	1	IC, 74LS08 quad AND gate
2008	1	IC, 7410 triple 3-input NAND
2029	3	IC, 7474 dual D flip-flop
2035	1	IC, 7493 4-bit binary counter
2042	1	IC, 74LS138 3-to-8 decoder
2083	2	IC, 74273 octal latch
2130	2	IC, 8T28B quad bi-bus driver (early version used P/N 2133, 8216: see Sec. 1.3 above)
2202	1	IC, 8008 central processing unit
3000	2	TRANSISTOR, 2N5138 PNP
3500	1	LED, red, with red frosted lens
3501	1	LED, red, with clear lens
4000-27	2	RESISTOR, 15 ohms, 10%, ¼w (color code brown, green, black)
4000-42	2	RESISTOR, 270 ohms, 10%, ¼w (color code red, violet, brown)
4000-43	2	RESISTOR, 330 ohms, 10%, ¼w (color code orange, orange, brown)
4000-55	7	RESISTOR, 3.3K, 10%, ¼w (color code orange, orange, red)
4000-61	1	RESISTOR, 10K, 10%, ¼w (color code brown, black, orange)
5501	9	CAPACITOR, .05 to 0.1 uf, 10 V min., ceramic
5502	1	CAPACITOR, 22 to 50 uf, 15 V min., tantalum
5900-2	1	CRYSTAL, 16.0 MHz

2.2 *8008-1 Option* The following parts are used with the 441-5, a version based on the premium 8008-1 (see Sec. 1.4).

Part No.	Quantity	Description
2034-2	1	IC, 7490 4-bit BCD counter, 16 MHz B input speed (replaces P/N 2035, 7493)
2203	1	IC, 8008-1, premium CPU (replaces P/N 2202, 8008)

2.3 *-12 V Power Option* The following part may be used to provide the 441 CPU board's -9 V, power requirement from a -12 V supply.

Part No.	Quantity	Description
3700-05	1	DIODE, IN5225 zener, 3.0 V, 5%, 500 mw

REV. A

SEC. 3 APPLICATION NOTES

3.1 *Parts Identification* A reproduction of the component side of the 441 CPU board appears in Fig. 3.1.1. The integrated circuits and other parts which make up the standard 441 are located as shown. These notes call out some of the options associated with the 441 board, for the convenience of the user.

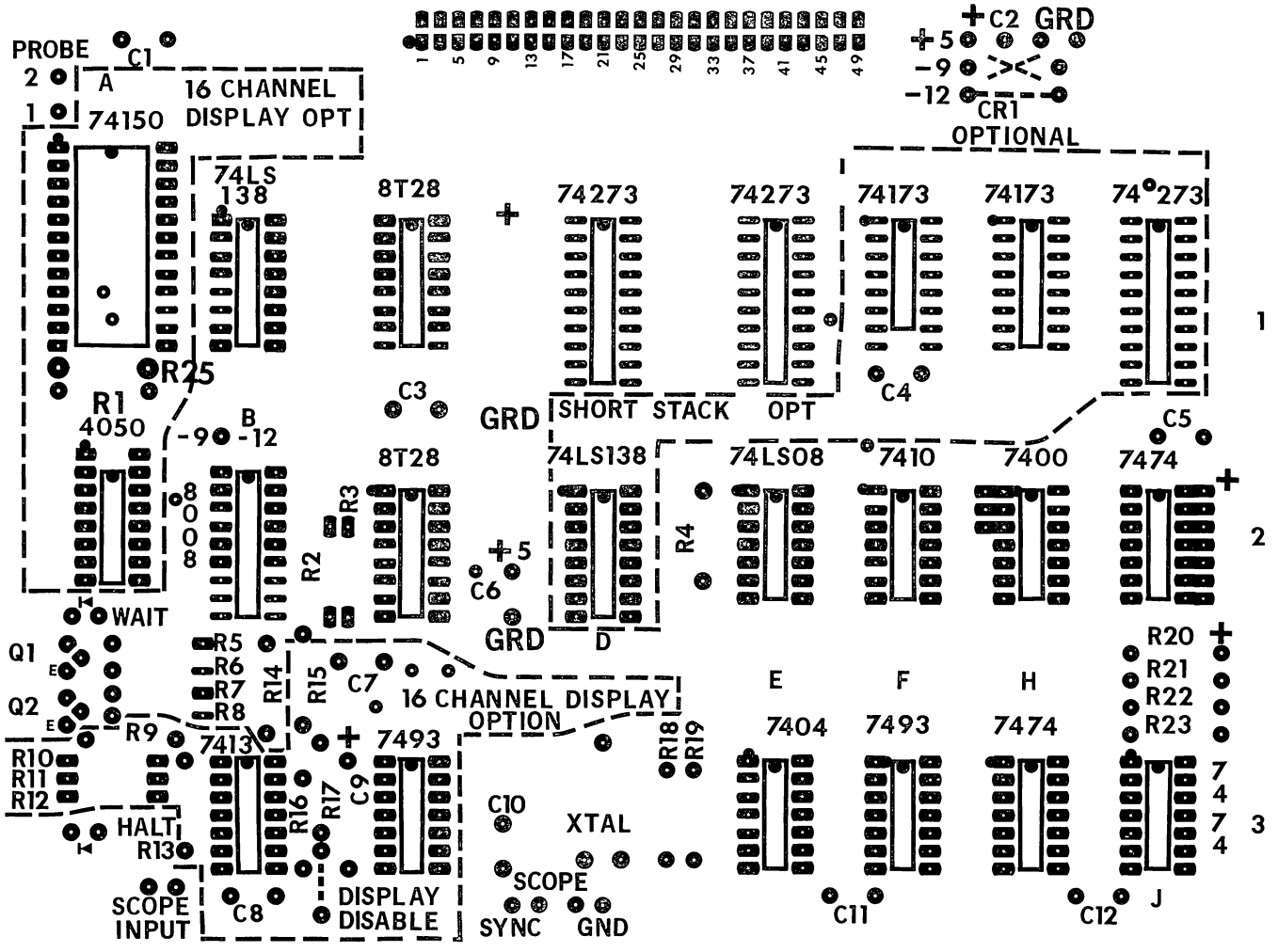


Figure 3.1.1--Component Side of Model 441 CPU Board

3.2 *16-Channel Scope Display Option* The Model 441 CPU board has space and all necessary foil wiring for the 16-Channel Scope Display, described briefly on the first page of this data sheet. A schematic and circuit description

modular micros martin research

SEC. 3 APPLICATION NOTES (cont'd)

are in the KT9007 data sheet. The locations of optional parts associated with the display are indicated in Fig. 3.1.1 with dotted lines, in two areas: near the left rear of the board, and left front of the board.

At the left rear are the two *PROBE* connections, *1* and *2,* which may be connected by the user to any signal of interest. Running along the front of the board are three pairs of holes, labeled *SCOPE INPUT, SCOPE SYNC,* and *GND.* By soldering in small loops of wire, with one end in each hole, the user provides convenient terminals for attachment of the oscilloscope probes.

When the display has been installed and powered up, it causes repetitive interrupts to the 8008. In order to execute more than the first few dozen instructions, it will be necessary to disable the display circuitry. This is done by connecting a jumper, as indicated by the dotted line at the left front of the board, labeled *DISPLAY DISABLE.*

3.3 *Short Stack Option* Both a circuit description and a schematic for the Short Stack option, KT9000, are available in a separate data sheet. This option consists of four integrated circuits (74LS138; 74273; 2-74173) whose locations are indicated by the irregular dotted area toward the right rear of the 441 CPU board.

3.4 *-12 V Option* The 8008 requires power supply voltages of +5 V and -9 V, while all other 441 circuitry requires +5 V only. If the user wishes to supply the -9 V from a -12 V supply, he should perform the following steps:

(1) Locate the two pairs of holes marked *-9* and *-12* near the top right of the board. The -9 pair has a dotted *X* in the middle.

(2) Using a knife or other sharp edge, cut through the foil wiring beneath the *X.* This disconnects the 8008 from the -9 V lines on the bus (pins 46 and 47).

(3) Install *CR1,* a 3.0 V zener diode, over the -12 V dotted line. A diode symbol appears in the foil on the 441 board, for orientation. If the zener is marked with a band or a +, install this marking to the right of the component side of the board.

This data sheet consists of reprints of literature published by the 8008 manufacturer, Intel Corporation. The 8008/8008-1 microprocessor is the heart of the central processing board of the MIKE 2, a microcomputer available from Martin Research. For replacement purposes, the 8008 is stocked as Part No. 2202, and the 8008-1 as P/N 2203.

SINGLE CHIP EIGHT-BIT PARALLEL CENTRAL PROCESSOR UNIT

- **Instruction Cycle Time — 12.5 μs with 8008-1 or 20 μs with 8008**
- **Directly addresses 16K x 8 bits of memory (RAM, ROM, or S.R.)**
- **Interrupt Capability**
- **48 Instructions, Data Oriented**
- **Address stack contains eight 14-bit registers (including program counter) which permit nesting of subroutines up to seven levels**

The 8008 is a single chip MOS 8-bit parallel central processor unit

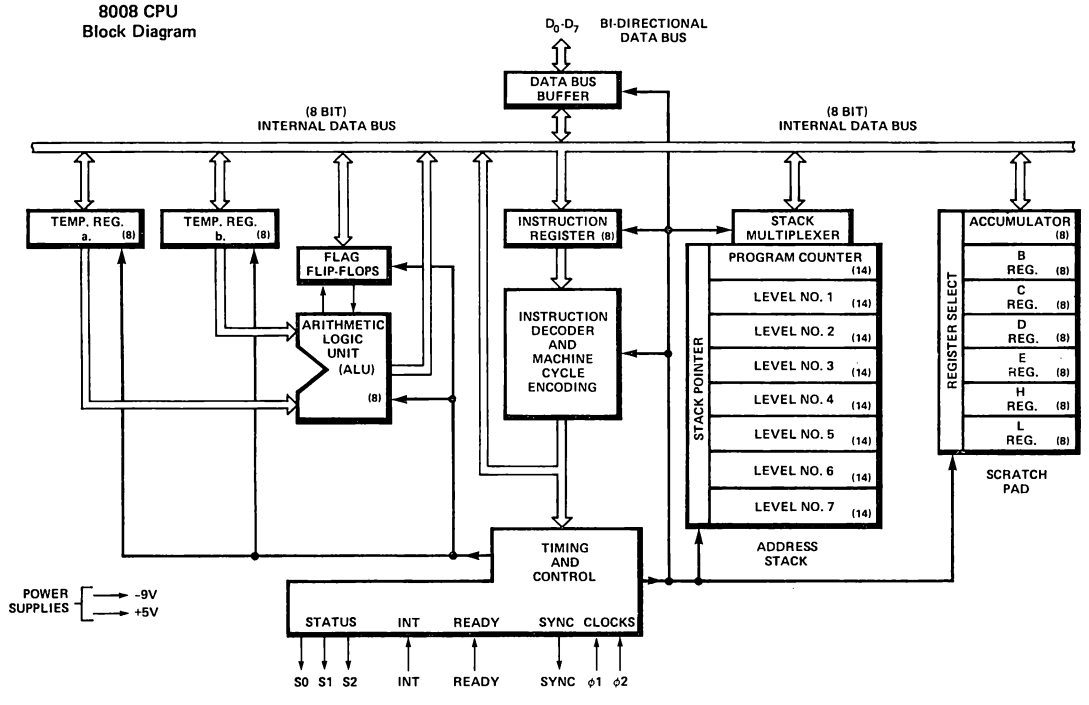
This CPU contains six 8-bit data registers, an 8-bit accumulator, two 8-bit temporary registers, four flag bits (carry, zero, sign, parity), and an 8-bit parallel binary arithmetic unit which implements addition, subtraction, and logical operations. A memory stack containing a 14-bit program counter and seven 14-bit words is used internally to store program and subroutine addresses. The 14-bit address permits the direct addressing of 16K words of memory (any mix of RAM, ROM or S.R.).

The instruction set of the 8008 consists of 48 instructions including data manipulation, binary arithmetic, and jump to subroutine.

The normal program flow of the 8008 may be interrupted through the use of the INTERRUPT control line. This allows the servicing of slow I/O peripheral devices while also executing the main program.

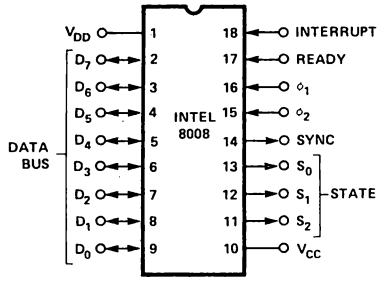
The READY command line synchronizes the 8008 to the memory cycle allowing any type or speed of semiconductor memory to be used.

8008 CPU
Block Diagram



modular micros martin research

8008 FUNCTIONAL PIN DESCRIPTION



D_0-D_7

BI-DIRECTIONAL DATA BUS. All address and data communication between the processor and the program memory, data memory, and I/O devices occurs on these 8 lines. Cycle control information is also available.

INT

INTERRUPT input. A logic "1" level at this input causes the processor to enter the INTERRUPT mode.

READY

READY input. This command line is used to synchronize the 8008 to the memory cycle allowing any speed memory to be used.

SYNC

SYNC output. Synchronization signal generated by the processor. It indicates the beginning of a machine cycle.

ϕ_1, ϕ_2

Two phase clock inputs.

S_0, S_1, S_2

MACHINE STATE OUTPUTS. The processor controls the use of the data bus and determines whether it will be sending or receiving data. State signals S_0, S_1, and S_2, along with SYNC inform the peripheral circuitry of the state of the processor.

V_{CC} +5V ±5%

V_{DD} -9V ±5%

COPYRIGHT NOTICE

The information in this data sheet has been reprinted with the permission of the copyright holder, Intel Corporation. Copyright © 1973, 1974, 1975, Intel Corp.

modular micros martin research

PROCESSOR TIMING

The 8008 is a complete central processing unit intended for use in any arithmetic, control, or decision-making system. The internal organization is centered around an 8-bit internal data bus. All communication within the processor and with external components occurs on this bus in the form of 8-bit bytes of address, instruction or data. (Refer to the accompanying block diagram for the relationship of all of the internal elements of the processor to each other and to the data bus.)

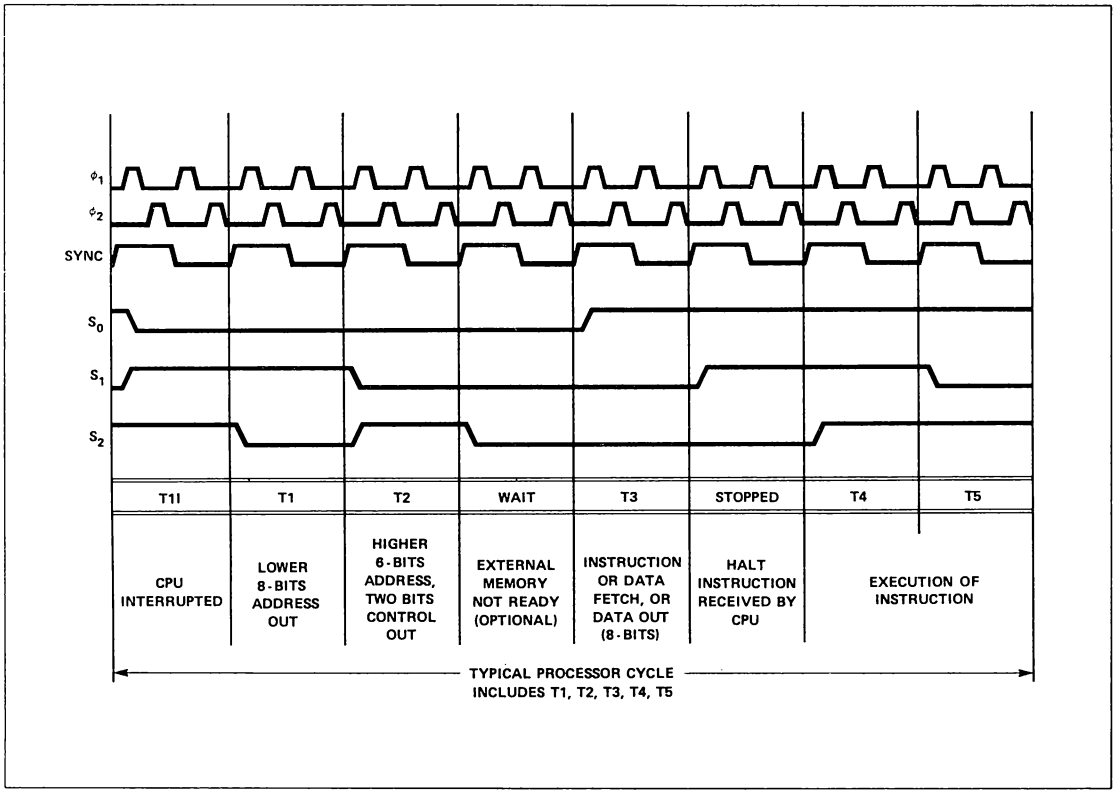
A. State Control Coding

The processor controls the use of the data bus and determines whether it will be sending or receiving data. State signals S_0, S_1, and S_2, along with SYNC inform the peripheral circuitry of the state of the processor. A table of the binary state codes and the designated state names is shown below.

S_0	S_1	S_2	STATE
0	1	0	T1
0	1	1	T1I
0	0	1	T2
0	0	0	WAIT
1	0	0	T3
1	1	0	STOPPED
1	1	1	T4
1	0	1	T5

B. Timing

Typically, a machine cycle consists of five states, two states in which an address is sent to memory (T1 and T2), one for the instruction or data fetch (T3), and two states for the execution of the instruction (T4 and T5). If the processor is used with slow memories, the READY line synchronizes the processor with the memories. When the memories are not available for either sending or receiving data, the processor goes into the WAIT state. The accompanying diagram illustrates the processor activity during a single cycle.



Basic 8008 Instruction Cycle

modular micros martin research

The receipt of an INTERRUPT is acknowledged by the T1I. When the processor has been interrupted, this state replaces T1. A READY is acknowledged by T3. The STOPPED state acknowledges the receipt of a HALT instruction.

Many of the instructions for the 8008 are multi-cycle and do not require the two execution states, T4 and T5. As a result, these states are omitted when they are not needed and the 8008 operates asynchronously with respect to the cycle length. The external state transition is shown below. Note that the WAIT state and the STOPPED may be indefinite in length (each of these states will be 2n clock periods). The use of READY and INTERRUPT with regard to these states will be explained later.

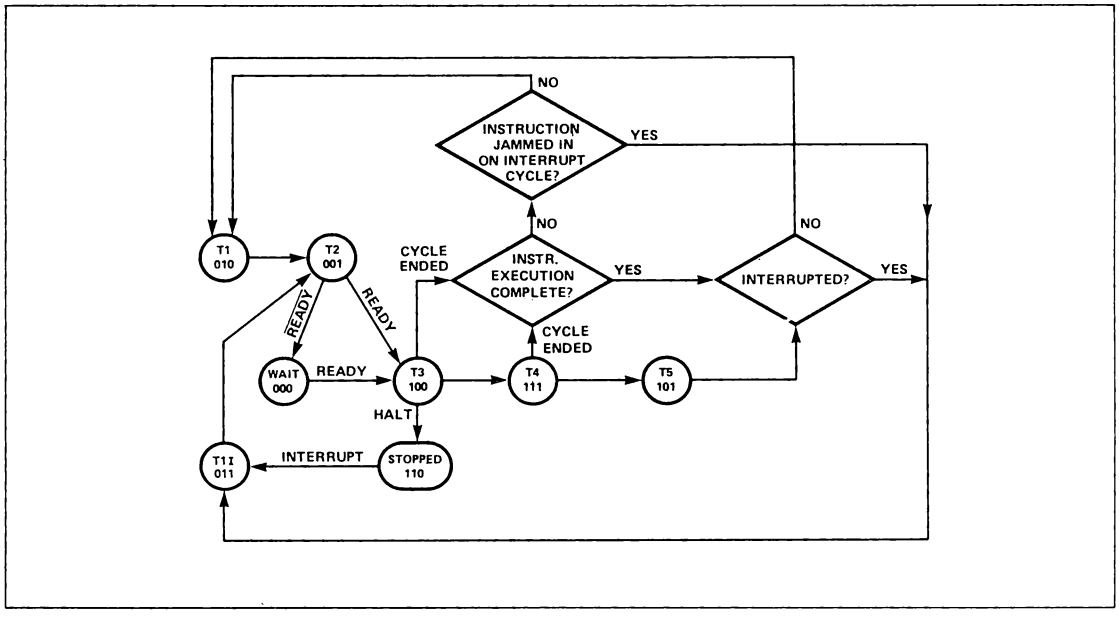


Figure 2. CPU State Transition Diagram

C. Cycle Control Coding

As previously noted, instructions for the 8008 require one, two, or three machine cycles for complete execution. The first cycle is always an instruction fetch cycle (PCI). The second and third cycles are for data reading (PCR), data writing (PCW), or I/O operations (PCC).

The cycle types are coded with two bits, D_6 and D_7, and are only present on the data bus during T2.

D_6	D_7	CYCLE	FUNCTION
0	0	PCI	Designates the address is for a memory read (first byte of instruction).
0	1	PCR	Designates the address is for a memory read data (additional bytes of instruction or data).
1	0	PCC	Designates the data as a command I/O operation.
1	1	PCW	Designates the address is for a memory write data.

modular micros martin research

BASIC FUNCTIONAL BLOCKS

The four basic functional blocks of this Intel processor are the instruction register, memory, arithmetic-logic unit, and I/O buffers. They communicate with each other over the internal 8-bit data bus.

A. Instruction Register and Control

The instruction register is the heart of all processor control. Instructions are fetched from memory, stored in the instruction register, and decoded for control of both the memories and the ALU. Since instruction executions do not all require the same number of states, the instruction decoder also controls the state transitions.

B. Memory

Two separate dynamic memories are used in the 8008, the pushdown address stack and a scratch pad. These internal memories are automatically refreshed by each WAIT, T3, and STOPPED state. In the worst case the memories are completely refreshed every eighty clock periods.

1. Address Stack

The address stack contains eight 14-bit registers providing storage for eight lower and six higher order address bits in each register. One register is used as the program counter (storing the effective address) and the other seven permit address storage for nesting of subroutines up to seven levels. The stack automatically stores the content of the program counter upon the execution of a CALL instruction and automatically restores the program counter upon the execution of a RETURN. The CALLs may be nested and the registers of the stack are used as last in/first out pushdown stack. A three-bit address pointer is used to designate the present location of the program counter. When the capacity of the stack is exceeded the address pointer recycles and the content of the lowest level register is destroyed. The program counter is incremented immediately after the lower order address bits are sent out. The higher order address bits are sent out at T2 and then incremented if a carry resulted from T1. The 14-bit program counter provides direct addressing of 16K bytes of memory. Through the use of an I/O instruction for bank switching, memory may be indefinitely expanded.

2. Scratch Pad Memory or Index Registers

The scratch pad contains the accumulator (A register) and six additional 8-bit registers (B, C, D, E, H, L). All arithmetic operations use the accumulator as one of the operands. All registers are independent and may be used for temporary storage. In the case of instructions which require operations with a register in external memory, scratch pad registers H & L provide indirect addressing capability; register L contains the eight lower order bits of address and register H contains the six higher order bits of address (in this case bit 6 and bit 7 are "don't cares").

C. Arithmetic/Logic Unit (ALU)

All arithmetic and logical operations (ADD, ADD with carry, SUBTRACT, SUBTRACT with borrow, AND, EXCLUSIVE OR, OR, COMPARE, INCREMENT, DECREMENT) are carried out in the 8-bit parallel arithmetic unit which includes carry-look-ahead logic. Two temporary resisters, register "a" and register "b", are used to store the accumulator and operand for ALU operations. In addition, they are used for temporary address and data storage during intra-processor transfers. Four control bits, carry flip-flop (c) , zero flip-flop (z) , sign flip-flop (s) , and parity flip-flop (p) , are set as the result of each arithmetic and logical operation. These bits provide conditional branching capability through CALL, JUMP, or RETURN on condition instructions. In addition, the carry bit provides the ability to do multiple precision binary arithmetic.

D. I/O Buffer

This buffer is the only link between the processor and the rest of the system. Each of the eight buffers is bi-directional and is under control of the instruction register and state timing. Each of the buffers is low power TTL compatible on the output and TTL compatible on the input.

modular micros martin research

BASIC INSTRUCTION SET

The following section presents the basic instruction set of the 8008.

A. Data and Instruction Formats

Data in the 8008 is stored in the form of 8-bit binary integers. All data transfers to the system data bus will be in the same format.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0

DATA WORD

The program instructions may be one, two, or three bytes in length. Multiple byte instructions must be stored in successive words in program memory. The instruction formats then depend on the particular operation executed.

		TYPICAL INSTRUCTIONS
One Byte Instructions		
D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	OP CODE	Register to register, memory reference, I/O arithmetic or logical, rotate or return instructions
Two Byte Instructions		
D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	OP CODE	
D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	OPERAND	Immediate mode instructions
Three Byte Instructions		
D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	OP CODE	
D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	LOW ADDRESS	JUMP or CALL instructions
X X D_5 D_4 D_3 D_2 D_1 D_0	HIGH ADDRESS*	*For the third byte of this instruction, D_6 and D_7 are "don't care" bits.

B. Summary of Processor Instructions

Index Register Instructions

The load instructions do not affect the flag flip-flops. The increment and decrement instructions affect all flip-flops except the carry.

MNEMONIC	MINIMUM STATES REQUIRED	INSTRUCTION CODE D_7 D_6	D_5 D_4 D_3	D_2 D_1 D_0	DESCRIPTION OF OPERATION
(1) Lr_1r_2	(5)	1 1	D D D	S S S	Load index register r_1 with the content of index register r_2.
(2) LrM	(8)	1 1	D D D	1 1 1	Load index register r with the content of memory register M.
LMr	(7)	1 1	1 1 1	S S S	Load memory register M with the content of index register r.
(3) Lrl	(8)	0 0 B B	D D D B B B	1 1 0 B B B	Load index register r with data B . . . B.
LMI	(9)	0 0 B B	1 1 1 B B B	1 1 0 B B B	Load memory register M with data B . . . B.
INr	(5)	0 0	D D D	0 0 0	Increment the content of index register r (r ≠ A).
DCr	(5)	0 0	D D D	0 0 1	Decrement the content of index register r (r ≠ A).

Accumulator Group Instructions

The result of the ALU instructions affect all of the flag flip-flops. The rotate instructions affect only the carry flip-flop.

ADr	(5)	1 0	0 0 0	S S S	Add the content of index register r, memory register M, or data B . . . B to the accumulator. An overflow (carry) sets the carry flip-flop.
ADM	(8)	1 0	0 0 0	1 1 1	
ADI	(8)	0 0 B B	0 0 0 B B B	1 0 0 B B B	
ACr	(5)	1 0	0 0 1	S S S	Add the content of index register r, memory register M, or data B . . . B to the accumulator with carry. An overflow (carry) sets the carry flip-flop.
ACM	(8)	1 0	0 0 1	1 1 1	
ACI	(8)	0 0 B B	0 0 1 B B B	1 0 0 B B B	
SUr	(5)	1 0	0 1 0	S S S	Subtract the content of index register r, memory register M, or data B . . . B from the accumulator. An underflow (borrow) sets the carry flip-flop.
SUM	(8)	1 0	0 1 0	1 1 1	
SUI	(8)	0 0 B B	0 1 0 B B B	1 0 0 B B B	
SBr	(5)	1 0	0 1 1	S S S	Subtract the content of index register r, memory register M, or data data B . . . B from the accumulator with borrow. An underflow (borrow) sets the carry flip-flop.
SBM	(8)	1 0	0 1 1	1 1 1	
SBI	(8)	0 0 B B	0 1 1 B B B	1 0 0 B B B	

modular micros martin research

MNEMONIC	MINIMUM STATES REQUIRED	INSTRUCTION CODE $D_7 D_6$	$D_5 D_4 D_3$	$D_2 D_1 D_0$	DESCRIPTION OF OPERATION
NDr	(5)	1 0	1 0 0	S S S	Compute the logical AND of the content of index register r, memory register M, or data B . . . B with the accumulator.
NDM	(8)	1 0	1 0 0	1 1 1	
NDI	(8)	0 0 B B	1 0 0 B B B	1 0 0 B B B	
XRr	(5)	1 0	1 0 1	S S S	Compute the EXCLUSIVE OR of the content of index register r, memory register M, or data B . . . B with the accumulator.
XRM	(8)	1 0	1 0 1	1 1 1	
XRI	(8)	0 0 B B	1 0 1 B B B	1 0 0 B B B	
ORr	(5)	1 0	1 1 0	S S S	Compute the INCLUSIVE OR of the content of index register r, memory register m, or data B . . . B with the accumulator.
ORM	(8)	1 0	1 1 0	1 1 1	
ORI	(8)	0 0 B B	1 1 0 B B B	1 0 0 B B B	
CPr	(5)	1 0	1 1 1	S S S	Compare the content of index register r, memory register M, or data B . . . B with the accumulator. The content of the accumulator is unchanged.
CPM	(8)	1 0	1 1 1	1 1 1	
CPI	(8)	0 0 B B	1 1 1 B B B	1 0 0 B B B	
RLC	(5)	0 0	0 0 0	0 1 0	Rotate the content of the accumulator left.
RRC	(5)	0 0	0 0 1	0 1 0	Rotate the content of the accumulator right.
RAL	(5)	0 0	0 1 0	0 1 0	Rotate the content of the accumulator left through the carry.
RAR	(5)	0 0	0 1 1	0 1 0	Rotate the content of the accumulator right through the carry.

Program Counter and Stack Control Instructions

(4) JMP	(11)	0 1 B_2 B_2 X X	X X X B_2 B_2 B_2 B_3 B_3 B_3	1 0 0 B_2 B_2 B_2 B_3 B_3 B_3	Unconditionally jump to memory address $B_3 \ldots B_3B_2 \ldots B_2$.
(5) JFc	(9 or 11)	0 1 B_2 B_2 X X	0 C_4 C_3 B_2 B_2 B_2 B_3 B_3 B_3	0 0 0 B_2 B_2 B_2 B_3 B_3 B_3	Jump to memory address $B_3 \ldots B_3B_2 \ldots B_2$ if the condition flip-flop c is false. Otherwise, execute the next instruction in sequence.
JTc	(9 or 11)	0 1 B_2 B_2 X X	1 C_4 C_3 B_2 B_2 B_2 B_3 B_3 B_3	0 0 0 B_2 B_2 B_2 B_3 B_3 B_3	Jump to memory address $B_3 \ldots B_3B_2 \ldots B_2$ if the condition flip-flop c is true. Otherwise, execute the next instruction in sequence.
CAL	(11)	0 1 B_2 B_2 X X	X X X B_2 B_2 B_2 B_3 B_3 B_3	1 1 0 B_2 B_2 B_2 B_3 B_3 B_3	Unconditionally call the subroutine at memory address $B_3 \ldots B_3B_2 \ldots B_2$. Save the current address (up one level in the stack).
CFc	(9 or 11)	0 1 B_2 B_2 X X	0 C_4 C_3 B_2 B_2 B_2 B_3 B_3 B_3	0 1 0 B_2 B_2 B_2 B_3 B_3 B_3	Call the subroutine at memory address $B_3 \ldots B_3B_2 \ldots B_2$ if the condition flip-flop c is false, and save the current address (up one level in the stack.) Otherwise, execute the next instruction in sequence.
CTc	(9 or 11)	0 1 B_2 B_2 X X	1 C_4 C_3 B_2 B_2 B_2 B_3 B_3 B_3	0 1 0 B_2 B_2 B_2 B_3 B_3 B_3	Call the subroutine at memory address $B_3 \ldots B_3B_2 \ldots B_2$ if the condition flip-flop c is true, and save the current address (up one level in the stack). Otherwise, execute the next instruction in sequence.
RET	(5)	0 0	X X X	1 1 1	Unconditionally return (down one level in the stack).
RFc	(3 or 5)	0 0	0 C_4 C_3	0 1 1	Return (down one level in the stack) if the condition flip-flop c is false. Otherwise, execute the next instruction in sequence.
RTc	(3 or 5)	0 0	1 C_4 C_3	0 1 1	Return (down one level in the stack) if the condition flip-flop c is true. Otherwise, execute the next instruction in sequence.
RST	(5)	0 0	A A A	1 0 1	Call the subroutine at memory address AAA000 (up one level in the stack).

Input/Output Instructions

INP	(8)	0 1	0 0 M	M M 1	Read the content of the selected input port (MMM) into the accumulator.
OUT	(6)	0 1	R R M	M M 1	Write the content of the accumulator into the selected output port (RRMMM, RR ≠ 00).

Machine Instruction

HLT	(4)	0 0	0 0 0	0 0 X	Enter the STOPPED state and remain there until interrupted.
HLT	(4)	1 1	1 1 1	1 1 1	Enter the STOPPED state and remain there until interrupted.

NOTES:
(1) SSS = Source Index Register, DDD = Destination Index Register. These registers, r_i, are designated A(accumulator—000), B(001), C(010), D(011), E(100), H(101), L(110).
(2) Memory registers are addressed by the contents of registers H & L.
(3) Additional bytes of instruction are designated by BBBBBBBB.
(4) X = "Don't Care".
(5) Flag flip-flops are defined by C_4C_3: carry (00-overflow or underflow), zero (01-result is zero), sign (10-MSB of result is "1"), parity (11-parity is even).

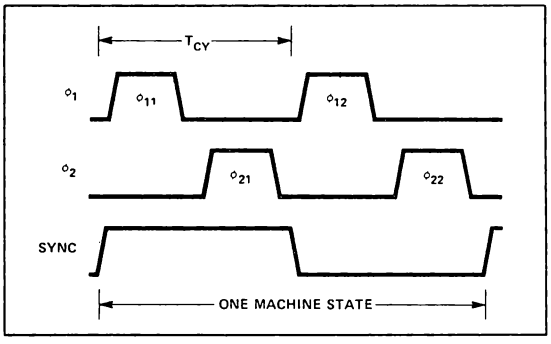
modular micros martin research

Internal Processor Operation

Internally the processor operates through five different states:

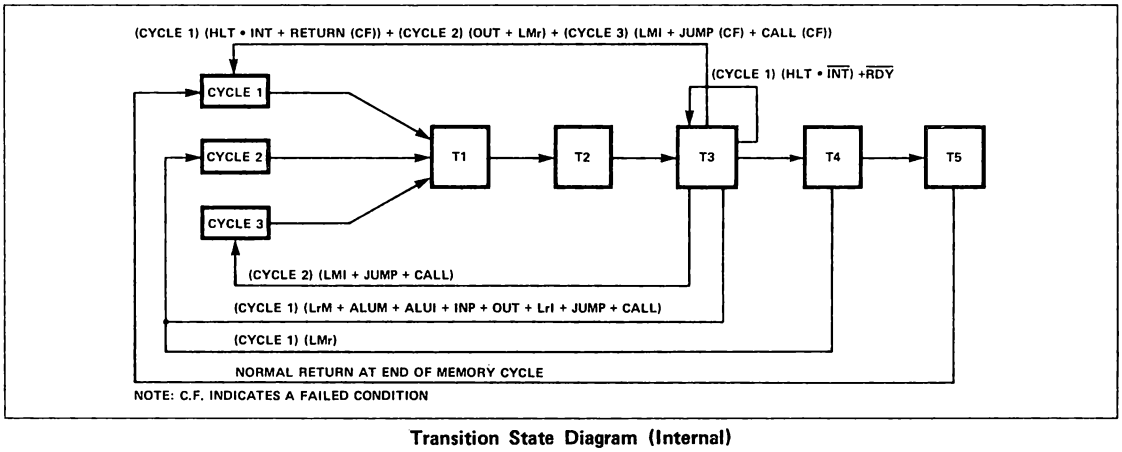
	Internal State	Typical Function
T1	NORMAL	Send out lower eight bits of address and increment program counter.
	INTERRUPT	Send out lower eight bits of address and suppress incrementing of program counter and acknowledge interrupt.
T2		Send out six higher order bits of address and two control bits, D_6 and D_7. Increment program counter if there has been a carry from T1.
T3	WAIT	Wait for READY signal to come true. Refresh internal dynamic memories while waiting.
	NORMAL	Fetch and decode instruction; fetch data from memory; output data to memory. Refresh internal memories.
	STOPPED	Remain stopped until INTERRUPT occurs. Refresh internal memories.
T4 and T5		Execute instruction and appropriately transfer data within processor. Content of data bus transfer is available at I/O bus for convenience in testing. Some cycles do not require these states. In those cases, the states are skipped and the processor goes directly to T1.

The 8008 is driven by two non-overlapping clocks. Two clock periods are required for each state of the processor. ϕ_1 is generally used to precharge all data lines and memories and ϕ_2 controls all data transfers within the processor. A SYNC signal (divide by two of ϕ_2) is sent out by the 8008. This signal distinguishes between the two clock periods of each state.



Processor Clocks

The figure below shows state transitions relative to the internal operation of the processor. As noted in the previous table, the processor skips unnecessary execution steps during any cycle. The state counter within the 8008 operates is a five bit feedback shift register with the feedback path controlled by the instruction being executed. When the processor is either waiting or stopped, it is internally cycling through the T3 state. This state is the only time in the cycle when the internal dynamic memories can be refreshed.


(CYCLE 1) (HLT • INT + RETURN (CF)) + (CYCLE 2) (OUT + LMr) + (CYCLE 3) (LMI + JUMP (CF) + CALL (CF))
(CYCLE 1) (HLT • $\overline{INT}$) +$\overline{RDY}$
CYCLE 1
CYCLE 2
CYCLE 3
T1
T2
T3
T4
T5
(CYCLE 2) (LMI + JUMP + CALL)
(CYCLE 1) (LrM + ALUM + ALUI + INP + OUT + LrI + JUMP + CALL)
(CYCLE 1) (LMr)
NORMAL RETURN AT END OF MEMORY CYCLE
NOTE: C.F. INDICATES A FAILED CONDITION

Transition State Diagram (Internal)

The following pages show the processor activity during each state of the execution of each instruction.

modular micros
martin research

INTERNAL PROCESSOR OPERATION

INDEX REGISTER INSTRUCTIONS

INSTRUCTION CODING				OPERATION	# OF STATES TO EXECUTE INSTRUCTION	MEMORY CYCLE ONE (1)					MEMORY CYCLE TWO					MEMORY CYCLE THREE				
$D_7 D_6$	$D_5 D_4 D_3$	$D_2 D_1 D_0$				T1 (2)	T2	T3	T4 (3)	T5	T1	T2	T3	T4	T5	T1	T2	T3	T4	T5
1 1	D D D	S S S		Lr_1r_2	(5)	PCL_{OUT} (4)	PCH_{OUT}	FETCH INSTR (5) TO IR & REG. b	SSS TO REG. b (6)	REG. b TO DDD										
1 1	D D D	1 1 1		LrM	(8)	PCL_{OUT}	PCH_{OUT}	FETCH INSTR. TO IR & REG. b	→ (7)		REG. L OUT (8)	REG. H OUT	DATA TO REG. b	X (9)	REG. b TO DDD					
1 1	1 1 1	S S S		LMr	(7)	PCL_{OUT}	PCH_{OUT}	FETCH INSTR. TO IR & REG. b	SSS TO REG. b	→	REG. L OUT (10)	REG. H OUT	REG. b TO OUT							
0 0	D D D	1 1 0		Lrl	(8)	PCL_{OUT}	PCH_{OUT}	FETCH INSTR. TO IR & REG. b	→		PCL_{OUT} (8)	PCH_{OUT}	DATA TO REG. b	X	REG. b TO DDD					
0 0	1 1 1	1 1 0		LMI	(9)	PCL_{OUT}	PCH_{OUT}	FETCH INSTR. TO IR & REG. b	→		PCL_{OUT} (8)	PCH_{OUT}	DATA TO REG. b	→		REG. L OUT (10)	REG. H OUT	REG. b TO OUT		
0 0	D D D	0 0 0		INr	(5)	PCL_{OUT}	PCH_{OUT}	FETCH INSTR. TO IR & REG. b	X	ADD OP - FLAGS AFFECTED										
0 0	D D D	0 0 1		DCr	(5)	PCL_{OUT}	PCH_{OUT}	FETCH INSTR. TO IR & REG. b	X	SUB OP - FLAGS AFFECTED										

ACCUMULATOR GROUP INSTRUCTIONS

1 0	P P P	S S S		ALU OP r	(5)	PCL_{OUT}	PCH_{OUT}	FETCH INSTR. TO IR & REG. b	SSS TO REG. b	ALU OP - FLAGS AFFECTED										
1 0	P P P	1 1 1		ALU OP M	(8)	PCL_{OUT}	PCH_{OUT}	FETCH INSTR. TO IR & REG. b	→		REG. L OUT (8)	REG. H OUT	DATA TO REG. b	X	ALU OP - FLAGS AFFECTED					
0 0	P P P	1 0 0		ALU OP I	(8)	PCL_{OUT}	PCH_{OUT}	FETCH INSTR. TO IR & REG. b	→		PCL_{OUT} (8)	PCH_{OUT}	DATA TO REG. b	X	ARITH OP - FLAGS AFFECTED					
0 0	0 0 0	0 1 0		RLC	(5)	PCL_{OUT}	PCH_{OUT}	FETCH INSTR. TO IR & REG. b	X	ROTATE REG. A CARRY AFFECTED										
0 0	0 0 1	0 1 0		RRC	(5)	PCL_{OUT}	PCH_{OUT}	FETCH INSTR. TO IR & REG. b	X	ROTATE REG. A CARRY AFFECTED										
0 0	0 1 0	0 1 0		RAL	(5)	PCL_{OUT}	PCH_{OUT}	FETCH INSTR. TO IR & REG. b	X	ROTATE REG. A CARRY AFFECTED										
0 0	0 1 1	0 1 0		RAR	(5)	PCL_{OUT}	PCH_{OUT}	FETCH INSTR. TO IR & REG. b	X	ROTATE REG. A CARRY AFFECTED										

PROGRAM COUNTER AND STACK CONTROL INSTRUCTIONS

0 1	X X X	1 0 0		JMP	(11)	PCL_{OUT}	PCH_{OUT}	FETCH INSTR. TO IR & REG. b	→		PCL_{OUT} (8)	PCH_{OUT}	LOWER ADD. TO REG. b	→		PCL_{OUT} (8)	PCH_{OUT}	HIGHER ADD. REG. a	REG. a TO PCH	REG. b TO PCL
0 1	U C C	0 0 0		JFc	(9 or 11)	PCL_{OUT}	PCH_{OUT}	FETCH INSTR. TO IR & REG. b	→		PCL_{OUT} (8)	PCH_{OUT}	LOWER ADD. TO REG. b	→		PCL_{OUT} (8)	PCH_{OUT}	HIGHER ADD. REG. a (11)	REG. a TO PCH	REG. b TO PCL
0 1	1 C C	0 0 0		JTc	(9 or 11)	PCL_{OUT}	PCH_{OUT}	FETCH INSTR. TO IR & REG. b	→		PCL_{OUT} (8)	PCH_{OUT}	LOWER ADD. TO REG. b	→		PCL_{OUT} (8)	PCH_{OUT}	HIGHER ADD. REG. a (11)	REG. a TO PCH	REG. b TO PCL
0 1	X X X	1 1 0		CAL	(11)	PCL_{OUT}	PCH_{OUT}	FETCH INSTR. TO IR & REG. b	→		PCL_{OUT} (8)	PCH_{OUT}	LOWER ADD. TO REG. b	→		PCL_{OUT} (8)	PCH_{OUT}	HIGHER ADD. REG. a	REG. a TO PCH	REG. b TO PCL
0 1	0 C C	0 1 0		CFc	(9 or 11)	PCL_{OUT}	PCH_{OUT}	FETCH INSTR. TO IR & REG. b	→		PCL_{OUT} (8)	PCH_{OUT}	LOWER ADD. TO REG. b	→		PCL_{OUT} (8)	PCH_{OUT}	HIGHER ADD. REG. a (12)	REG. a TO PCH	REG. b TO PCL
0 1	1 C C	0 1 0		CTc	(9 or 11)	PCL_{OUT}	PCH_{OUT}	FETCH INSTR. TO IR & REG. b	→		PCL_{OUT} (8)	PCH_{OUT}	LOWER ADD. TO REG. b	→		PCL_{OUT} (8)	PCH_{OUT}	HIGHER ADD. REG. a (12)	REG. a TO PCH	REG. b TO PCL
0 0	X X X	1 1 1		RET	(5)	PCL_{OUT}	PCH_{OUT}	FETCH INSTR. TO IR & REG. b	POP STACK	X										
0 0	0 C C	0 1 1		RFc	(3 or 5)	PCL_{OUT}	PCH_{OUT}	FETCH INSTR. TO IR & REG. b	POP STACK (13)	X										
0 0	1 C C	0 1 1		RTc	(3 or 5)	PCL_{OUT}	PCH_{OUT}	FETCH INSTR. TO IR & REG. b	POP STACK (13)	X										
0 0	A A A	1 0 1		RST	(5)	PCL_{OUT}	PCH_{OUT}	FETCH INSTR. TO REG. b AND PUSH STACK (0→REG. a)	REG. a TO PCH	REG. b TO PCL (14)										

I/O INSTRUCTIONS

0 1	0 0 M	M M 1		INP	(8)	PCL_{OUT}	PCH_{OUT}	FETCH INSTR. TO IR & REG. b	→		REG. A TO OUT (15)	REG. b TO OUT	DATA TO REG. b	COND FF OUT (16)	REG. b TO REG. A					
0 1	R R M	M M 1		OUT	(6)	PCL_{OUT}	PCH_{OUT}	FETCH INSTR. TO IR & REG. b	→		REG. A TO OUT (15)	REG. b TO OUT	X (17)							

MACHINE INSTRUCTIONS

0 0	0 0 0	0 0 X		HLT	(4)	PCL_{OUT}	PCH_{OUT}	FETCH INSTR. TO IR & REG. b & HALT (18)												
1 1	1 1 1	1 1 1		HLT	(4)	PCL_{OUT}	PCH_{OUT}	FETCH INSTR. TO IR & REG. b & HALT (18)												

NOTES

1. The first memory cycle is always a PCI (instruction) cycle.
2. Internally, states are defined as T1 through T5. In some cases more than one memory cycle is required to execute an instruction.
3. Content of the internal data bus at T4 and T5 is available at the data bus. This is designed for testing purposes only.
4. Lower order address bits in the program counter are denoted by PCL and higher order bits are designated by PCH.
5. During an instruction fetch the instruction comes from memory to the instruction register and is decoded.
6. Temporary registers are used internally for arithmetic operations and data transfers (Register a and Register b.)
7. These states are skipped.
8. PCR cycle (Memory Read Cycle).
9. "X" denotes an idle state.
10. PCW cycle (Memory Write Cycle)
11. When the JUMP is conditional and the condition fails, states T4 and T5 are skipped and the state counter advances to the next memory cycle.
12. When the CALL is conditional and the condition fails, states T4 and T5 are skipped and the state counter advances to the next memory cycle. If the condition is true, the stack is pushed at T4, and the lower and higher order address bytes are loaded into the program counter.
13. When the RETURN condition is true, pop up the stack, otherwise, advance to next memory cycle skipping T4 and T5.
14. Bits D_3 through D_5 are loaded into PCL and all other bits are set to zero, zeros are loaded into PCH.
15. PCC cycle (I/O Cycle).
16. The content of the condition flip-flops is available at the data bus S at D_0, Z at D_1, P at D_2, C at D_3.
17. A READY command must be supplied for the OUT operation to be completed. An idle T3 state is used and then the state counter advances to the next memory cycle.
18. When a HALT command occurs, the CPU internally remains in the T3 state until an INTERRUPT is recognized. Externally, the STOPPED state is indicated.

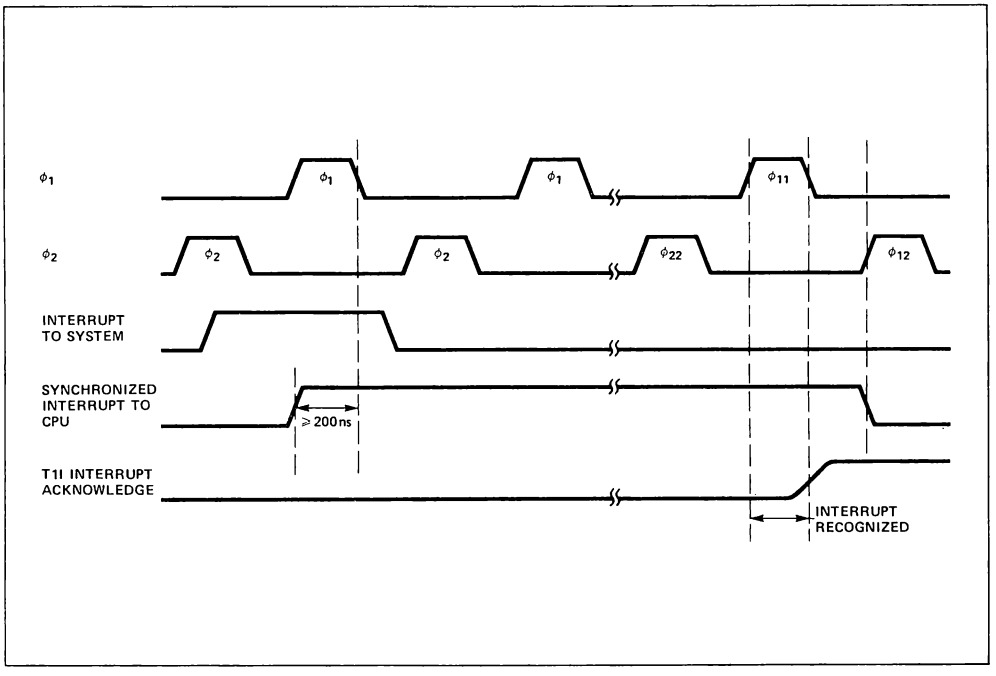
PROCESSOR CONTROL SIGNALS

A. Interrupt Signal (INT)

1) INTERRUPT REQUEST

If the interrupt line is enabled (Logic "1"), the CPU recognizes an interrupt request at the next instruction fetch (PCI) cycle by outputting $S_0 S_1 S_2$ = 011 at T1I time. The lower and higher order address bytes of the program counter are sent out, but the program counter is not advanced. A successive instruction fetch cycle can be used to insert an arbitrary instruction into the instruction register in the CPU. (If a multi-cycle or multi-byte instruction is inserted, an interrupt need only be inserted for the first cycle.)

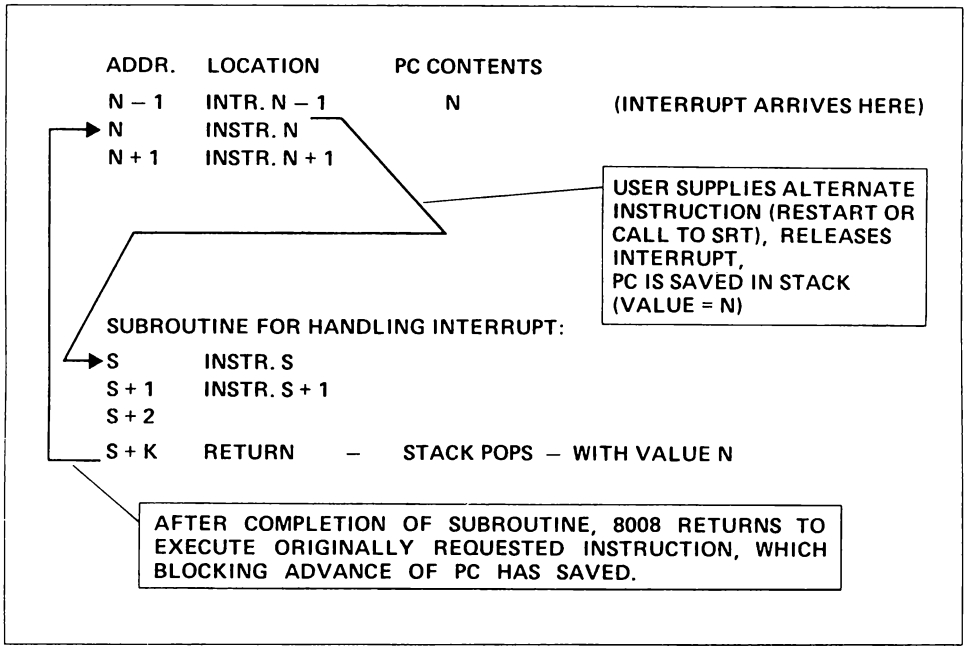
When the processor is interrupted, the system INTERRUPT signal must be synchronized with the leading edge of the ϕ_1 or ϕ_2 clock. To assure proper operation of the system, the interrupt line to the CPU must not be allowed to change within 200ns of the falling edge of ϕ_1.



Recognition of Interrupt

If a HALT is inserted, the CPU enters a STOPPED state; if a NOP is inserted, the CPU continues; if a "JUMP to 0" is inserted, the processor executes program from location 0, etc. The RESTART instruction is particularly useful for handling interrupt routines since it is a one byte call.

modular micros martin research



8008 Interrupt

2) START-UP OF THE 8008

When power (V_{DD}) and clocks (ϕ_1, ϕ_2) are first turned on, a flip-flop internal to the 8008 is set by sensing the rise of V_{DD}. This internal signal forces a HALT (00000000) into the instruction register and the 8008 is then in the STOPPED state. The following sixteen clock periods after entering the STOPPED state are required to clear (logic "0") memories (accumulator, scratch pad, program counter, and stack). During this time the interrupt line has been at logic "0". Any time after the memories are cleared, the 8008 is ready for normal operation.

To reset the flip-flop and also escape from the stopped state, the interrupt line must go to a logic "1"; It should be returned to logic "0" by decoding the state T1I at some time later than ϕ_{11}. Note that whenever the 8008 is in a T1I state, the program counter is not incremented. As a result, the same address is sent out on two successive cycles.

Three possible sequences for starting the 8008 are shown on the following page. The RESTART instruction is effectively a one cycle call instruction, and it is convenient to use this instruction to call an initiation subroutine. Note that it is not necessary to start the 8008 with a RESTART instruction.

The selection of initiation technique to use depends on the sophistication of the system using the 8008. If the interrupt feature is used only for the start-up of the 8008 use the ROM directly, no additional external logic associated with instructions from source other than the ROM program need be considered. If the interrupt feature is used to jam instructions into the 8008, it would then be consistent to use it to jam the initial instruction.

The timing for the interrupt with the start-up timing is shown on an accompanying sheet. The jamming of an instruction and the suppression of the program counter update are handled the same for all interrupts.

modular micros martin research

EXAMPLE 1:

Shown below are two start-up alternatives where an instruction is not forced into the 8008 during the interrupt cycle. The normal program flow starts the 8008.

a.

8008 ADDRESS OUT	INSTRUCTION IN ROM		
0 0 0 0 0 0 0 0 0 0 0 0 0 0	NOP	(LAA 11 000 000)	Entry Directly To Main Program
0 0 0 0 0 0 0 0 0 0 0 0 0 0	NOP		
0 0 0 0 0 0 0 0 0 0 0 0 0 1	$INSTR_1$		
0 0 0 0 0 0 0 0 0 0 0 0 1 0	$INSTR_2$		

b.

8008 ADDRESS OUT	INSTRUCTION IN ROM		
0 0 0 0 0 0 0 0 0 0 0 0 0 0	RST	(RST = 00 XYZ 101)	A Jump To The Main Program
0 0 0 0 0 0 0 0 X Y Z 0 0 0	$INSTR_1$		
0 0 0 0 0 0 0 0 X Y Z 0 0 1	$INSTR_2$		
.	.		
.	.		
.	.		

EXAMPLE 2:

A RESTART instruction is jammed in and first instruction in ROM initially ignored.

8008 ADDRESS OUT	INSTRUCTION IN ROM		
0 0 0 0 0 0 0 0 0 0 0 0 0 0	$INSTR_1$	(RST = 00 XYZ 101)	Start-up Routine
0 0 0 0 0 0 0 0 X Y Z 0 0 0	$INSTR_a$		
0 0 0 0 0 0 0 0 X Y Z 0 0 1	$INSTR_b$		
.	.		
.	.		
.	.		
0 0 0 0 0 0 0 0 n n n n n n	RETURN		
0 0 0 0 0 0 0 0 0 0 0 0 0 0	$INSTR_1$	($INSTR_1$ executed now)	Main Program
0 0 0 0 0 0 0 0 0 0 0 0 0 1	$INSTR_2$		
.	.		
.	.		
.	.		

Note that during the interrupt cycle the flow of the instruction to the 8008 either from ROM or another source must be controlled by hardware external to 8008.

START-UP OF THE 8008

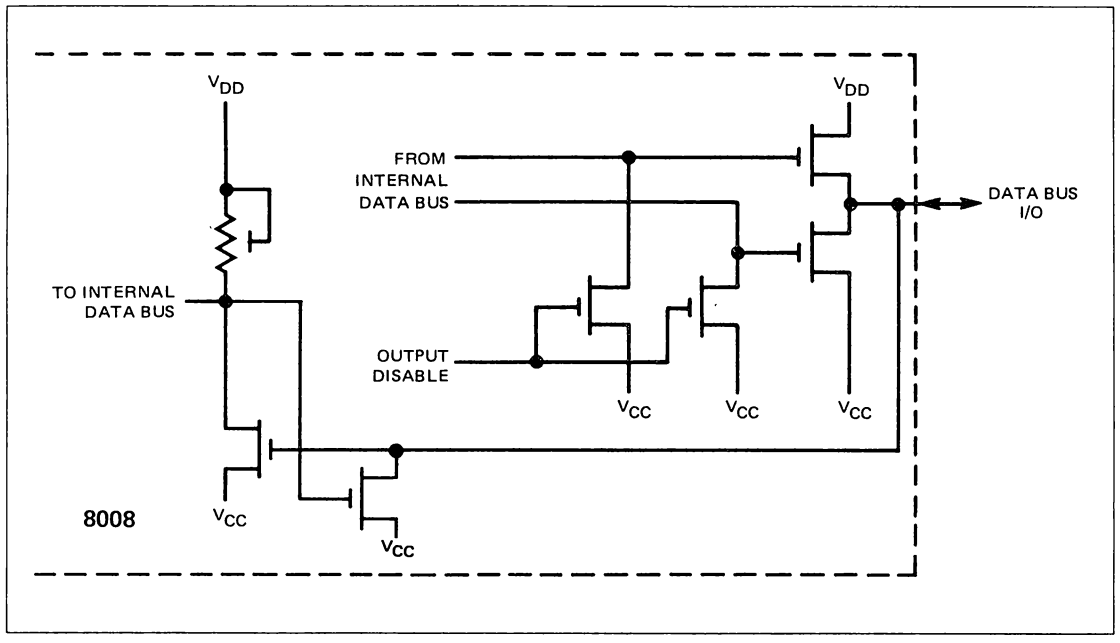
B. Ready (RDY)

The 8008 is designed to operate with any type or speed of semiconductor memory. This flexibility is provided by the READY command line. A high-speed memory will always be ready with data (tie READY line to V_{CC}) almost immediately after the second byte of the address has been sent out. As a result the 8008 will never be required to wait for the memory. On the other hand, with slow ROMs, RAMs or shift registers, the data will not be immediately available; the 8008 must wait until the READY command indicates that the valid memory data is available. As a result any type or any combination of memory types may be used. The READY command line synchronizes the 8008 to the memory cycle. **When a program is being developed, the READY signal provides a means of stepping through the program, one cycle at a time.**

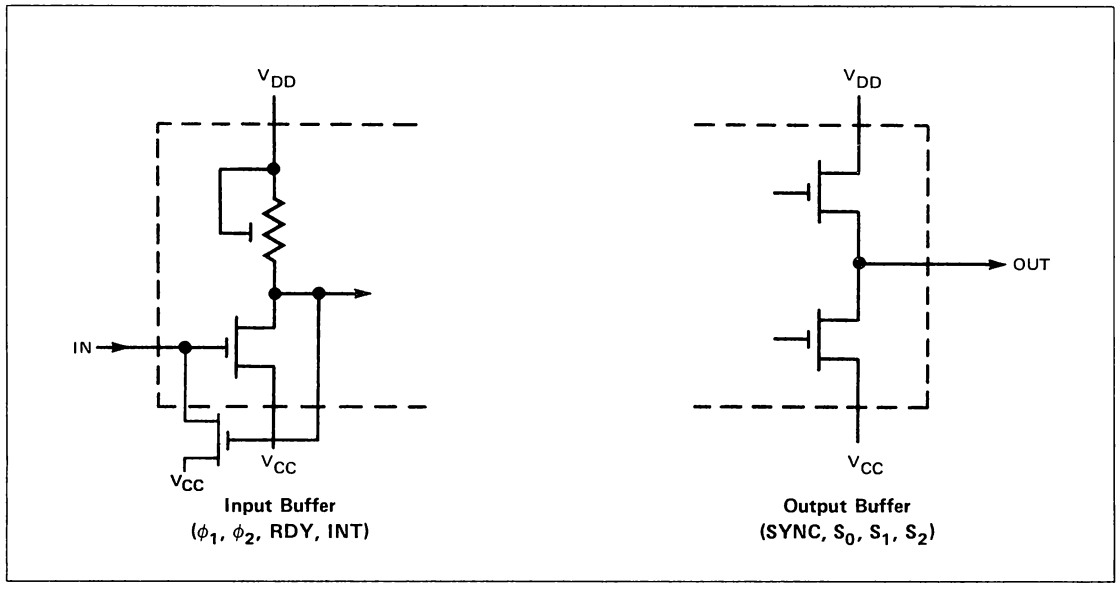
modular micros martin research

ELECTRICAL SPECIFICATIONS

The following pages provide the electrical characteristics for the 8008. All of the inputs are TTL compatible, but input pull-up resistors are recommended to insure proper V_{IH} levels. All outputs are low-power TTL compatible. The transfer of data to and from the data bus is controlled by the CPU. During both the WAIT and STOPPED states the data bus output buffers are disabled and the data bus is floating.



Data Bus I/O Buffer



I/O Circuitry

modular micros martin research

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias	0°C to +70°C
Storage Temperature	−55°C to +150°C
Input Voltages and Supply Voltage With Respect to V_{CC}	+0.5 to −20V
Power Dissipation	1.0 W @ 25°C

*COMMENT

Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other condition above those indicated in the operational sections of this specification is not implied.

D.C. AND OPERATING CHARACTERISTICS

T_A = 0°C to 70°C, V_{CC} = +5V ±5%, V_{DD} = −9V ±5% unless otherwise specified. Logic "1" is defined as the more positive level (V_{IH}, V_{OH}). Logic "0" is defined as the more negative level (V_{IL}, V_{OL}).

SYMBOL	PARAMETER	LIMITS MIN.	LIMITS TYP.	LIMITS MAX.	UNIT	TEST CONDITIONS
I_{DD}	AVERAGE SUPPLY CURRENT–OUTPUTS LOADED*		30	60	mA	T_A = 25°C
I_{LI}	INPUT LEAKAGE CURRENT			10	μA	V_{IN} = 0V
V_{IL}	INPUT LOW VOLTAGE (INCLUDING CLOCKS)	V_{DD}		V_{CC}−4.2	V	
V_{IH}	INPUT HIGH VOLTAGE (INCLUDING CLOCKS)	V_{CC}−1.5		V_{CC}+0.3	V	
V_{OL}	OUTPUT LOW VOLTAGE			0.4	V	I_{OL} = 0.44mA C_L = 200 pF
V_{OH}	OUTPUT HIGH VOLTAGE	V_{CC}−1.5			V	I_{OH} = 0.2mA

*Measurements are made while the 8008 is executing a typical sequence of instructions. The test load is selected such that at V_{OL} = 0.4V, I_{OL} = 0.44mA on each output.

A.C. CHARACTERISTICS

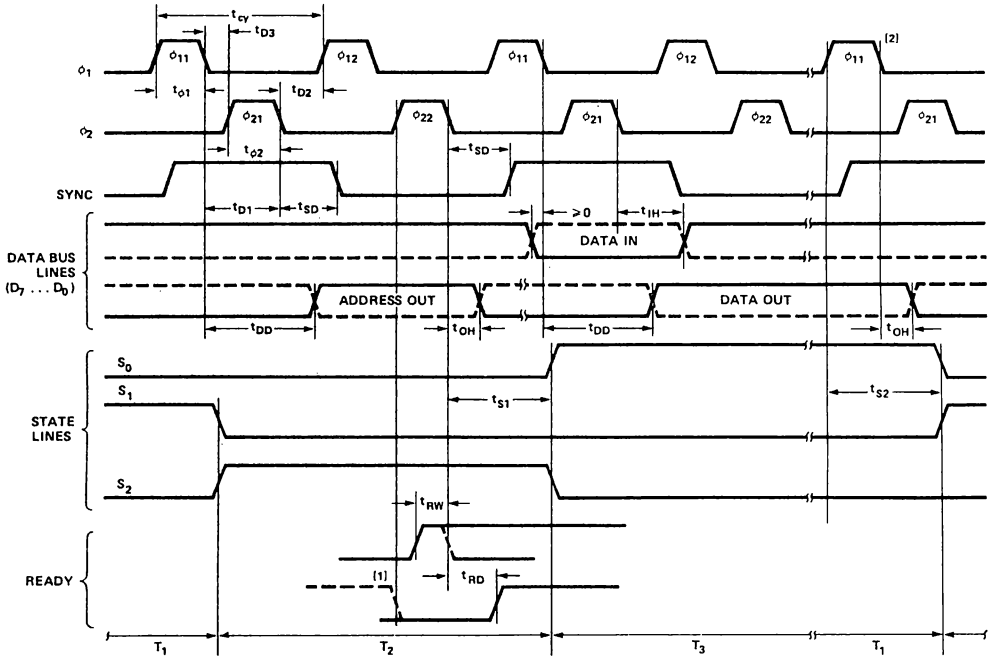
T_A = 0°C to 70°C; V_{CC} = +5V ±5%, V_{DD} = −9V ±5%. All measurements are referenced to 1.5V levels.

SYMBOL	PARAMETER	8008 LIMITS MIN.	8008 LIMITS MAX.	8008-1 LIMITS MIN.	8008-1 LIMITS MAX.	UNIT	TEST CONDITIONS
t_{CY}	CLOCK PERIOD	2	3	1.25	3	μs	t_R,t_F = 50ns
t_R,t_F	CLOCK RISE AND FALL TIMES		50		50	ns	
$t_{\phi 1}$	PULSE WIDTH OF ϕ_1	.70		.35		μs	
$t_{\phi 2}$	PULSE WIDTH OF ϕ_2	.55		.35		μs	
t_{D1}	CLOCK DELAY FROM FALLING EDGE OF ϕ_1 TO FALLING EDGE OF ϕ_2	.90	1.1		1.1	μs	
t_{D2}	CLOCK DELAY FROM ϕ_2 TO ϕ_1	.40		.35		μs	
t_{D3}	CLOCK DELAY FROM ϕ_1 TO ϕ_2	.20		.20		μs	
t_{DD}	DATA OUT DELAY		1.0		1.0	μs	C_L = 100pF
t_{OH}	HOLD TIME FOR DATA BUS OUT	.10		.10		μs	
t_{IH}	HOLD TIME FOR DATA IN	[1]		[1]		μs	
t_{SD}	SYNC OUT DELAY		.70		.70	μs	C_L = 100pF
t_{S1}	STATE OUT DELAY (ALL STATES EXCEPT T1 AND T1I)[2]		1.1		1.1	μs	C_L = 100pF
t_{S2}	STATE OUT DELAY (STATES T1 AND T1I)		1.0		1.0	μs	C_L = 100pF
t_{RW}	PULSE WIDTH OF READY DURING ϕ_{22} TO ENTER T3 STATE	.35		.35		μs	
t_{RD}	READY DELAY TO ENTER WAIT STATE	.20		.20		μs	

[1] t_{IH} MIN ≥ t_{SD} [2] If the INTERRUPT is not used, all states have the same output delay, t_{S1}.

modular micros martin research

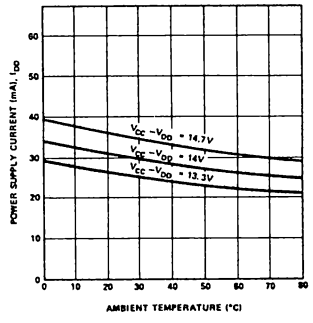
TIMING DIAGRAM



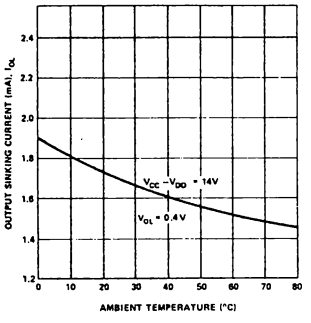
Notes: 1. READY line must be at "0" prior to ϕ_{22} of T_2 to guarantee entry into the WAIT state.
2. INTERRUPT line must not change levels within 200ns (max.) of falling edge of ϕ_1.

TYPICAL D.C. CHARACTERISTICS

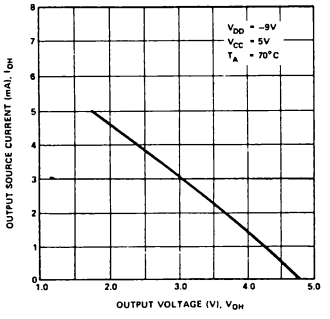
POWER SUPPLY CURRENT VS. TEMPERATURE



OUTPUT SINKING CURRENT VS. TEMPERATURE

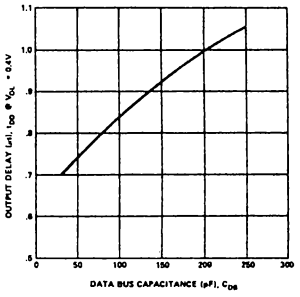


OUTPUT SOURCE CURRENT VS. OUTPUT VOLTAGE



TYPICAL A.C. CHARACTERISTICS

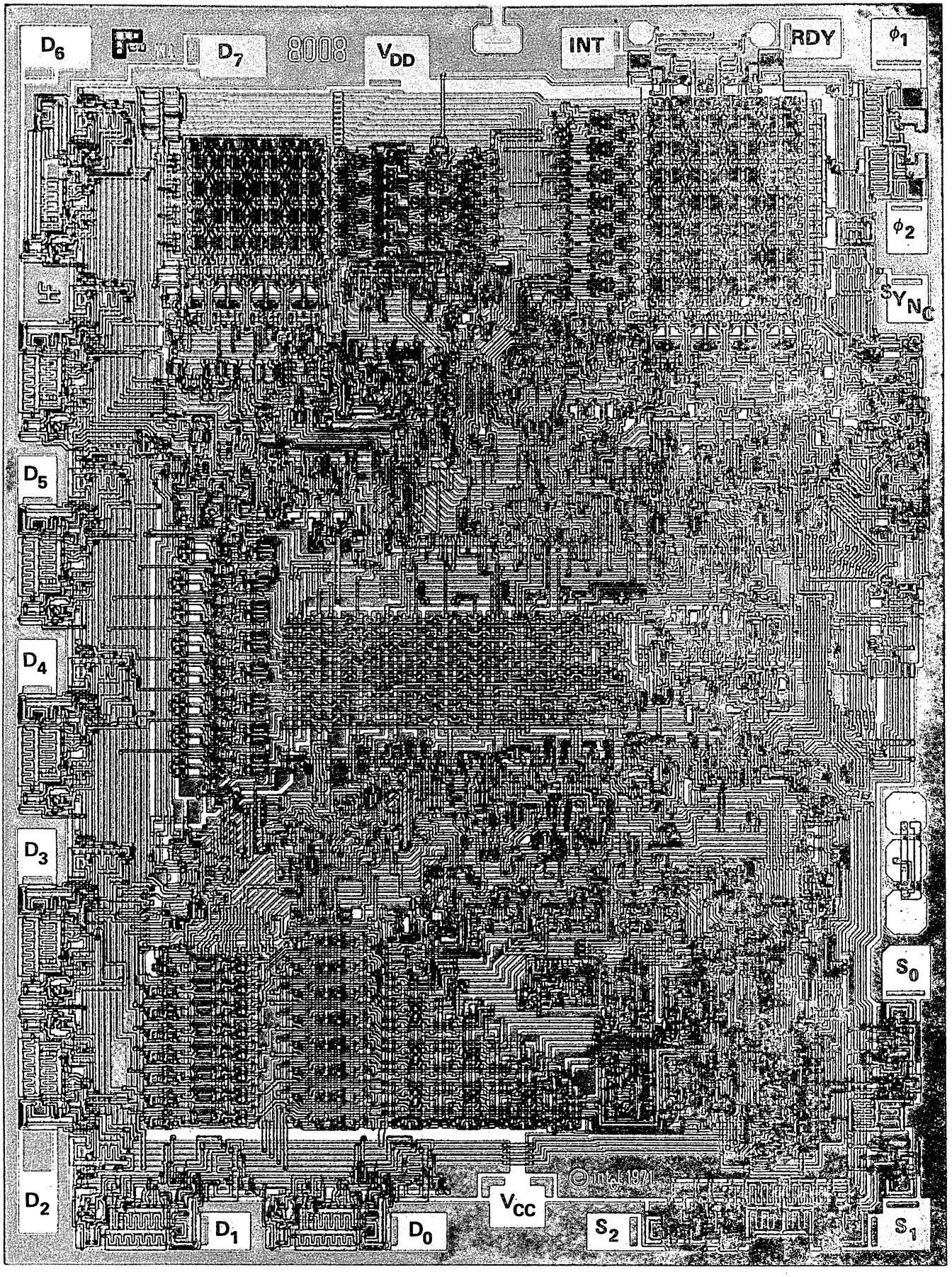
DATA OUT DELAY VS. OUTPUT LOAD CAPACITANCE



CAPACITANCE f = 1MHz; T_A = 25°C; Unmeasured Pins Grounded

SYMBOL	TEST	LIMIT (pF)	
		TYP.	MAX.
C_{IN}	INPUT CAPACITANCE	5	10
C_{DB}	DATA BUS I/O CAPACITANCE	5	10
C_{OUT}	OUTPUT CAPACITANCE	5	10

modular micros martin research



8008 Photomicrograph With Pin Designations

The *SHORT STACK* option consists of a kit of four integrated circuits which boost the interrupt-handling ability of the 8008 microprocessor. In conjunction with the Central Processing Unit and the *MIKE 2* Monitor Program, the Short Stack allows the *MIKE 2* user to save registers and flags as an interrupt begins, and to restore status later, allowing the main line program to continue without disruption.

SPECIFICATIONS

Interfacing — The Short Stack interfaces to the *modular micro* bus as a combined input/output port, addressed as Input 5. Both space and complete printed circuit wiring are provided on the Model 441 CPU board. However, the Short Stack may also be assembled at any other location on the microcomputer bus.

Software Required — For interrupt handling, a software register- and flag-saving routine is required. Suitable programming is included in the *MIKE 2* Monitor.

Memory Required — The *MIKE 2* Monitor requires both the two-byte temporary storage provided by the Short Stack, and page 013 in RAM, to save status during interrupts. Since minimal *MIKE 2* systems are sold without page 013, the Short Stack also includes the necessary RAM chips.

TECHNICAL DATA

SEC. 1 CIRCUIT DESCRIPTION

1.1 *Introduction: Why the Short Stack is Needed* — It is desirable for a computer to be able to service an interrupt request at any random point during the execution of the main line program. At that point, the 8008's internal registers and flags might have any of the possible combinations of values. If the main line program is to proceed normally when the interrupt ends, the interrupt subroutine must *save status* as it begins, and *restore* status before returning control to the main line program.

The 8008 has seven general-purpose index registers (A, B, C, D, E, H, and L). In addition, four internal flip-flops store *flag* bits--set during the execution of arithmetic and logic instructions. These flags are *Carry, Parity, Zero,* and *Sign.* The 8008 data sheet indicates how these flags are set and tested by various instructions. Obviously, to serve any useful purpose the interrupt routine must use some of these registers and flags. But if it fails to restore them to their original conditions, the main line program will soon be affected.

The usual method for saving status is to start the interrupt routine with instructions which successively write the contents of memory into designated locations in RAM. The states of the flags are tested and written into RAM as well. As the interrupt routine ends, the flags and registers are read back from memory and the original status is restored. However, without additional

SEC. 1 CIRCUIT DESCRIPTION (cont'd)

hardware, an 8008 microcomputer cannot save all seven registers with its instruction set. Unassisted, the 8008 can save only five. This is because the status-saving program must use the H and L registers for pointing to the location in RAM where registers are to be saved--which destroys the original contents of these two registers. The H and L register contents could be loaded into two other registers temporarily--such as the D and E--but then *those* contents are destroyed. The purpose of the Short Stack option is to provide a convenient place for the *MIKE 2* Monitor to store two eight-bit words temporarily, permitting the microcomputer to save all seven registers plus flags during an interrupt.

1.2 *Combined I/O Port* The Short Stack consists of two registers, for a storage capacity of two eight-bit data words. Register 1 is a 74273, an octal latch with normal TTL outputs. Register 2 is composed of a pair of 74173s, four-bit latches with three-state bus driver outputs. Both registers are addressed by a single 8008 instruction, INP 005 (Input 5), which serves as a combined I/O instruction. Each time this instruction is executed, the following data transfers take place:

(1) The contents of the 8008 A register are moved into Short Stack Register 1 (the 74273).

(2) The data word previously in Short Stack Register 1 is moved into Short Stack Register 2 (the 74173s).

(3) The data previously in Short Stack Register 2 is returned to the 8008 A register, where it remains as the Input 5 instruction ends.

Information transfers with the CPU take place via the 8008 A register--there is no necessity to address memory using the H and L registers. These features allow suitable software--such as that found in the *MIKE 2* Monitor--to save 8008 registers and flags during interrupt handling.

1.3 *System Timing* The Input 5 strobe, $\overline{\text{DN5}}$, is developed in the usual manner by a 74LS138/3205 strobe decoder. The $\overline{\text{DN5}}$ signal goes low at PCC-T3A time of the Input 5 instruction. A little later, at PCC-ST3 time, the strobe ends, going high again. Short Stack timing consists of two steps: *Phase A* occurs when $\overline{\text{DN5}}$ goes low; *Phase B* occurs when the strobe returns to the high level.

(A) *Register 2 to CPU*--When $\overline{\text{DN5}}$ goes low, the 74173 output enable (*OE*) pins are activated. The 74173 outputs leave the third state, and the Register 2 contents are driven onto the CPU data bus. This is PCC-T3A time, and the 8008 receives this input information. When the instruction is over, this data will reside in the 8008 A register.

(B) *CPU to Register 1*--As $\overline{\text{DN5}}$ ends, the 74273 is clocked on the 0-to-1 transition of this strobe. Information on the microcomputer's DL bus is latched up in Register 1. Here the Input 5 instruction doubles as an output instruction, one very useful feature of the 8008. The contents

©1975, Martin Research. Pp. 1-5, Rev. A, Nov. 1975.

SEC. 1 CIRCUIT DESCRIPTION (cont'd)

of the A register, set up by a previous instruction, were moved to the DL register at PCC-T1 time of the input/output instruction. This output data is still valid on the DL bus when the $\overline{DN5}$ strobe takes place (being destroyed only at the first T1 time of the following instruction).

Register 1 to Register 2--The 74173s are also clocked on the 0-to-1 transition of $\overline{DN5}$. Information on the output terminals of Register 1 is thus latched up into Register 2. Proper data transfer is assured, even though Registers 2 and 1 are clocked simulatenously. This is because Register 2's input terminals safely latch up the old information on the Register 1 output terminals before new information has had a chance to travel through the 74273. (The 74173 data enable terminals--$\overline{DE}$--are grounded, to prevent disabling the clock. The reset function is unused, and this terminal--R--is also grounded.)

SEC. 2 PARTS LIST

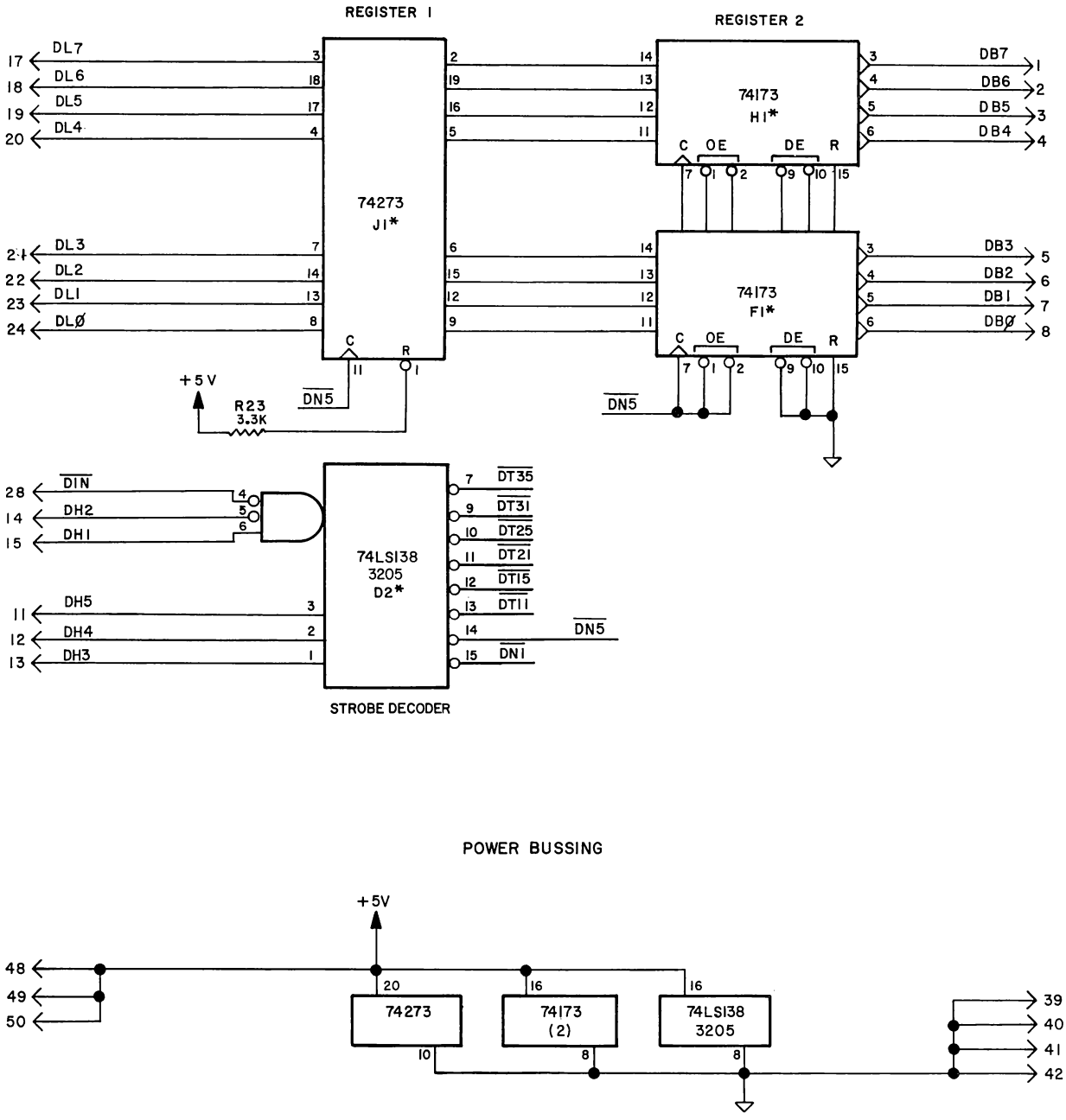
2.1 *Basic Short Stack* Available as KT9000-1, the Short Stack may be assembled either in the spaces provided on the Model 441 CPU board, or on the system breadboard. The kit also includes RAM chips in order to provide the system with page 013 in RAM. (See Part 3 for assembly.)

Part No.	Quantity	Description
1013	5	SOCKET, 16-pin DIP, solder-tail
1015	1	SOCKET, 20-pin DIP, solder-tail
1053	3	SOCKET, 16-pin DIP, wirewrap
1055	1	SOCKET, 20-pin DIP, wirewrap
2042	1	IC, 74LS138 3-to-8 decoder
2083	1	IC, 74273 octal latch
2105	2	IC, 74173 4-bit latch, tri-state
2205	2	IC, 2112 256 x 4 RAM, 1.0 us
4000-55	1	RESISTOR, 3.3K, 10%, ¼w

SEC. 3 INSTALLATION NOTES

3.1 *Four ICs on 441 CPU Board* The four integrated circuits shown in the KT9000 schematic--one 74LS138/3205; one 74273; two 74173s--can be installed easily in the 441 CPU board. The locations are indicated both on the 441 board itself, and by coordinate number on the KT9000 schematic (e.g., *J1*). The user should install solder-tail sockets at the appropriate locations, and simply plug in the ICs. The wirewrap sockets are unneeded and may be saved for use on the breadboard. *Notice:* when the Short Stack is to be assembled on the 441, it is unnecessary to install the 3.3K resistor shown on the schematic; it is already on the 441 board, as R23.

REV
A



*IC LOCATIONS, RESISTOR PART NO. REFER TO 441 CPU BD

NOTE: PROPRIETARY INFORMATION. NOT TO BE DISCLOSED WITHOUT EXPRESS WRITTEN PERMISSION.	DRAWN BY Schneider	CHECKED BY D.T.K	DRAWING NO. SD-9000 (Page 4)
	DATE 10-17-75	DATE 12/1/75	

EF4A

modular micros martin research

REV. A

SEC. 3 INSTALLATION NOTES (cont'd)

3.2 *Four ICs on a Breadboard* Alternately the Short Stack may be assembled on the system breadboard. This is the recommended procedure for users who have the 401 CPU board. In that case, the wirewrap sockets provided may be used in order to facilitate the connections necessary. Location of the chips is at the user's option. The user should be sure to install the 3.3K resistor shown on the schematic.

3.3 *Extra RAMs* Two 2112 RAMs are provided, with 16-pin solder-type sockets. These should be installed as page 013 in RAM, which must be present for proper status saving. For details on location on the PROM/RAM board, see the appropriate data sheet.

The *16-CHANNEL SCOPE DISPLAY* allows important microcomputer data and timing to be displayed on an external single-trace, triggered-sweep oscilloscope. Timing diagrams of important microcomputer system signals appear directly on the face of the scope. This display is useful both for debugging microcomputer hardware, and for educational purposes--since the user actually sees how data transfers take place.

The Model 9007 16-Channel Scope Display has been designed to interface with an 8008 CPU board. Its parts may conveniently be inserted into the Model 441 processor circuit, which has all necessary foil wiring and silkscreened parts outlines.

SPECIFICATIONS

Interfacing

This circuit is most readily assembled directly on the CPU board, because it displays signals which do not appear on the system bus. Fourteen of the display's sixteen channels are connected to the CPU data bus and timing signals; the two remaining channels are assigned to probes for tracing other system signals. The vertical and sweep inputs of a single-trace, triggered-sweep scope connect to test points in the 16-Channel Scope Display circuitry.

Instructions displayed

The Scope Display shows data transfers which occur during the first several dozen instructions in the system's Monitor program, beginning with location 000 000. The Display includes an oscillator which repetitively interrupts the 8008, causing it to restart from that location.

Power

The Scope Display derives its +5 volt power requirements from the CPU board on which it is assembled.

TECHNICAL DATA

SEC. 1 CIRCUIT DESCRIPTION

1.1 *Interrupt Oscillator* The Scope Display circuitry includes an interrupt oscillator which generates a continuous series of pulses, repetitively interrupting the 8008 microprocessor. The $\overline{INTD}$ line shown in the schematic diagram is connected to the $\overline{SET}$ terminal of a 7474 flip-flop in the CPU's interrupt logic circuitry. This connection is shown on the schematic diagram for the 441 CPU board, in the *Interrupt Logic* section.

The interrupt oscillator consists of one half of a 7413 Schmitt trigger NAND gate, which has a feedback resistor R16 between its input and inverting output. Assuming the input initially low, this causes the output to go high--which level, fed back, causes the input to go high in response. Now the output goes low again--and oscillation is established. A capacitor, C9, delays the feedback, and the circuit oscillates with a frequency dependent on the RC time constant. The output signal is differentiated by another RC combination, R1 and C8, to produce a shorter pulse, avoiding multiple interrupts. A CMOS buffer interfaces the RC circuit to the next TTL stage.

A jumper to ground at the 7413 input disables the oscillator, to allow normal execution of programs when testing is completed.

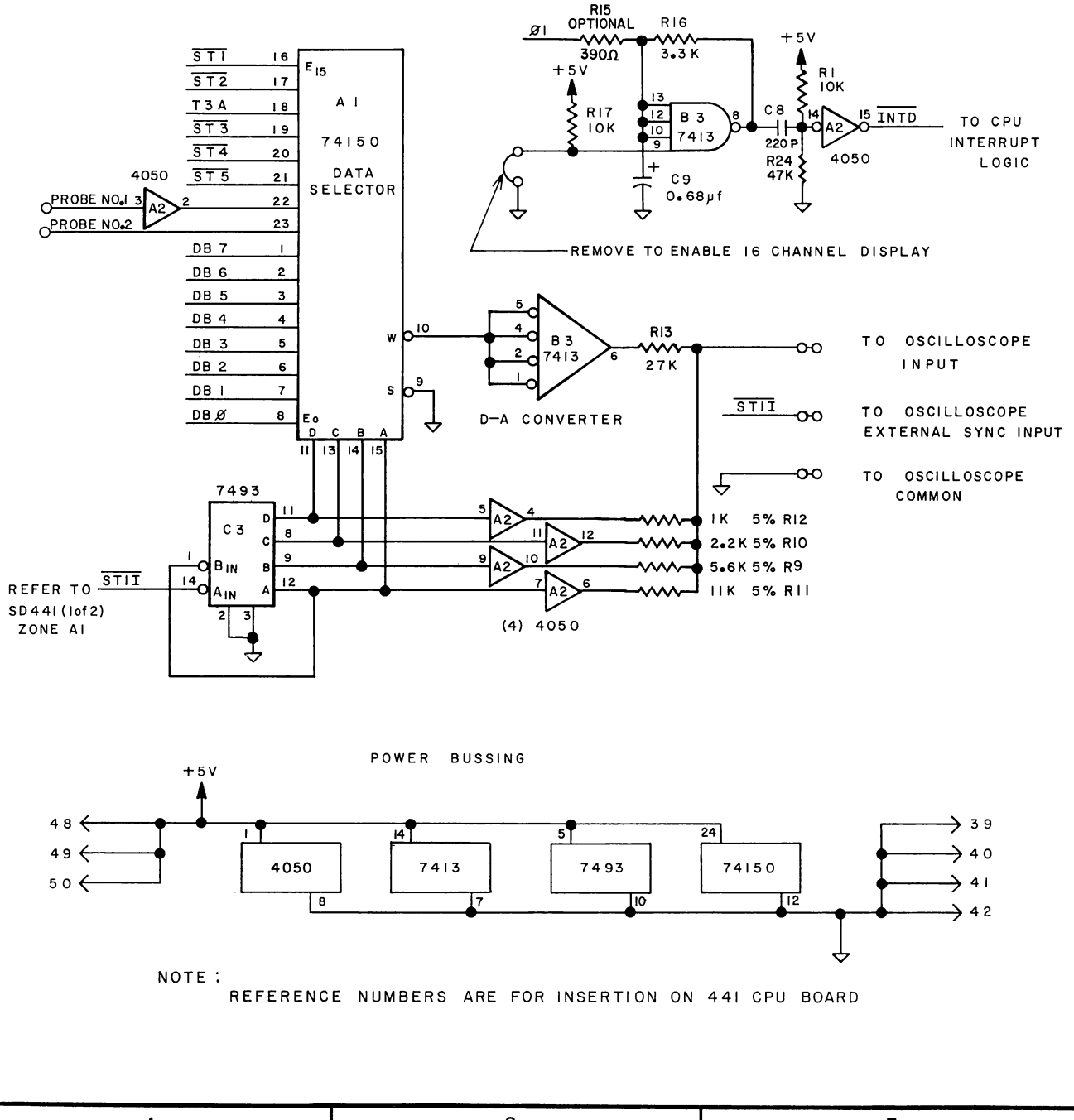
NOTE: R16, a 3.3K resistor connecting the interrupt oscillator to $\phi 1$, is optional. Intended to synchronize the interrupts with the CPU, to avoid an internal 8008 interrupt race, this resistor is unnecessary when using the Display on the 441 board. This is because the 441 interrupt logic includes a flip-flop synchronized with $\phi 2$. Only if this flip-flop (J3) were replaced with a jumper would it be necessary to use the R16 resistor.

1.2 *Channel Counter* As the 8008 recognizes an interrupt, it generates a brief negative pulse called $\overline{STII}$. (The source of this signal is shown on the 441 schematic diagram.) Each successive interrupt clocks a 7493 counter, which counts up one step, starting at 0 and proceeding through 15, and repeating indefinitely. The 7493's four output lines contain the four-bit binary code for the channel number being visualized at any given instant.

1.3 *Data Selector* The channel counter drives a 74150 data selector, which is essentially a single-pole, sixteen-throw switch. The four-bit channel code determines which one of its sixteen inputs is connected to its single output terminal. Thus, each time the CPU is interrupted, the data selector moves up one channel number. Since the 74150 output is inverted, one half of a 7413 is used to reinvert the signal.

©1975 Martin Research. Pp. 1-5, Rev. A, Nov. 1975.

REV
A



NOTE: PROPRIETARY INFORMATION. NOT TO BE DISCLOSED WITHOUT EXPRESS WRITTEN PERMISSION.

DRAWN BY Schneider	CHECKED BY DM	DRAWING NO. DS 9007 (Page 3)
DATE 11-19-75	DATE 11-19-75	

EF4A

modular micros
martin research

SEC. 1 CIRCUIT DESCRIPTION (cont'd)

1.4 *D/A Converter* The channel counter also drives four CMOS buffers, which normalize the 7493 TTL output voltages (which may range anywhere between about 2½ to 5 volts) to more consistent levels. The buffer outputs are connected through resistors to form a simple five-bit digital-to-analog converter.

The four most significant bits derive from the channel counter. The high-order counter bit (the 7493 D output) drives the lowest-valued resistor, which draws the most current and most strongly influences the output voltage to the scope. Each of the four lines tends to pull the common output connection up to +5 V, if it is high, or down to ground, if it is a logic zero. The resulting analog voltage determines the relative position of the selected channel on the screen. The values of the resistors R9 through R12 are not exactly in the normal binary progression (1K, 2K, 4K, 8K), but are chosen to display four groups of four lines each on the screen, for easier trace location. The 5% resolution of these resistors is more than adequate for pleasing display appearance.

The fifth, low-order bit derives from the signal that has been selected by the data selector. The relatively small current through R13 moves the trace up slightly when the signal is a logic one, and down slightly when the signal is a zero. Thus the scope draws a timing diagram for the selected channel, left to right, until the next interrupt occurs. Then the trace for the next channel starts at the left-hand side of the scope, slightly above the previous trace. The Display moves through the same microcomputer instruction sequence each time, switching so rapidly that the eye perceives a sixteen-channel pattern on the scope.

SEC. 2 PARTS LIST

2.1 *Standard Parts* The following parts are included with the KT9007, a kit for adding the 16-Channel Scope Display to the Model 441 CPU board.

Part No.	Quantity	Description
1012	2	SOCKET, 14-pin DIP, solder tail, tin
1013	1	SOCKET, 16-pin DIP, solder tail, tin
1017	1	SOCKET, 24-pin DIP, solder tail, tin
2010	1	IC, 7413 dual 4-input NAND, Schmitt trigger
2035	1	IC, 7493 4-bit binary counter
2044	1	IC, 74150 16-to-1 multiplexer
2420	1	IC, 4050 hex buffer, CMOS
4000-44	1	RESISTOR, 390 ohms, 10%, ¼W (color code orange, white, brown)
4000-61	2	RESISTOR, 10K, 10%, ¼W (color code brown, black, orange)
4000-66	1	RESISTOR, 27K, 10%, ¼W (color code red, violet, orange)

modular micros martin research

SEC. 2 PARTS LIST (cont'd)

Part No.	Quantity	Description
4001-97	1	RESISTOR, 1K, 5%, ¼W (color code brown, black, red, gold)
4001-105	1	RESISTOR, 2.2K, 5%, ¼W (color code red, red, red, gold)
4001-115	1	RESISTOR, 5.6K, 5%, ¼W (color code green, blue, red, gold)
4001-122	1	RESISTOR, 11K, 5%, ¼W (color code brown, brown, orange, gold)
5000-16	1	CAPACITOR, 220 pf, 1000 V, ceramic
5010	1	CAPACITOR, 0.68 uf, 20%, 15 V, tantalum

2.2 *Optional Part* The following part is optional (see Sec. 1.1 above):

Part No.	Quantity	Description
4000-55	1	RESISTOR, 3.3K, 10%, ¼ W (color code orange, orange, red)

SEC. 3 APPLICATION NOTES

3.1 *Use of Probes* Channels 7 and 8 on the 16-Channel Scope Display have been left open so that the user can visualize any desired signal on the screen. Probes numbers 2 and 1 are brought out at the left rear of the component side of the 441 CPU board. Probe No. 1 is connected to the data selector through a CMOS buffer, and is especially useful for connecting to a noisy signal line or to high-impedance circuitry which might be overloaded by Probe No. 2, an unbuffered TTL input. For example, the 8008's MOS circuitry could be overloaded under certain conditions by connecting an additional TTL load.

The channels which have been hardwired into the data selector on the 441 CPU board are made up of six important timing signals at the top, and the eight data bus leads on the bottom of the screen.

3.2 *Interrupt Period* With the parts shown, the trace begins again for every ten to fifteen 8008 instructions, approximately. If it is desired to visualize more of the Monitor program, the period can be stretched by connecting an additional capacitor in parallel with the 0.68 uf oscillator capacitor, C9.

3.3 *Scope Connections* The connections to the scope are at the left front of the component side of the 441 CPU board, and are identified clearly by the silkscreened legends. Each of the three terminals--SCOPE INPUT, SCOPE SYNC, and GND--has two adjacent holes. By soldering in a small loop of wire, with one end in each hole, the user can provide convenient terminals for attachment of the oscilloscope probes.

The *403 PROM/RAM BOARD* is a basic memory component for computer systems in the *modular micro* family. With room for up to 2K bytes of programmable read-only memory (PROM) and 1K bytes of random access memory (RAM), the 403 supports a system monitor program, and provides space for storage of users' programs.

SPECIFICATIONS

Memory Capacity	Up to 2K bytes of PROM: uses up to eight type 1702A 256 by 8 reprogrammable memory devices. Up to 1K of RAM: up to eight type 2112 256 by 4 static RAM chips.
Speed	RAM, 800 ns select time; PROM, 900 ns select time (meets requirements of 8008-1 system without memory wait cycles). Faster memory chips available as options.
Interfacing	Connects directly to the *modular micro* bus with standard fifty-pin connectors. Bus drivers on board ensure adequate drive power for expanded systems. PROMs occupy pages 000 through 007, RAMs page 010 through 013.
Power	With 256 bytes each of PROM and RAM, requires +5 V, 380 MA; -9 V, 35 MA; regulated ±5%.

TECHNICAL DATA

SEC. 1 CIRCUIT DESCRIPTION

1.1 *2112 RAMS* The 403 includes capacity for eight 2112 RAM chips, occupying pages 010 through 013 in memory. Each of these devices includes 1,024 memory cells, organized as 256 four-bit words. To interface with the eight-bit *modular micro* structure, the 2112s are combined into pairs, with one 2112 connected to the four high-order data bus lines on the 403 board (BB7 through BB4), and the other 2112 to the low-order lines (BB3 through BBO). Each pair makes up a *page* in memory--that is, 256 eight-bit words (bytes).

Since each 2112 pair has 256, or 2^8, memory words, eight address lines are required to select which word in a given page is to be written into or read from. Word selection is made by the *modular micro* low-order address bus, DL7 through DLO.

A given *page* in RAM is selected when the chip enable ($\overline{CE}$) terminal on the appropriate pair of 2112s is brought low. In the 403 schematic, the page 013 chip enable signal is called $\overline{CE13}$; there are four such signals, $\overline{CE10}$ through $\overline{CE13}$, corresponding to memory pages 010 through 013.

SEC. 1 CIRCUIT DESCRIPTION (cont'd)

The four data input/output terminals of each 2112 RAM use three-state bidirectional bus-driving circuitry. When a 2112 is not chip-enabled, its data terminals are in the high-impedance state, and neither sink nor source any appreciable amount of current. No data transfers can occur between the 2112's memory cells and the bus; some other device on the bus determines bus line voltage levels.

When a 2112 is chip-enabled, its input/output bus drivers are activated. The direction in which they drive is determined by the state of the write enable ($\overline{WE}$) terminal. If this signal is high, the 2112 is in the READ mode; the bus drivers point outwards; and memory data can be read into the CPU. If $\overline{WE}$ is low, the 2112 is in the WRITE mode; the bus drivers point inwards; and the CPU can write data into the selected memory word.

1.1 *1702A PROMs* The 403 also includes room for eight 1702A PROMs, occupying pages 000 through 007 in memory. Each PROM includes 2,048 memory cells, organized as 256 eight-bit words. The 1702A has twice the capacity of the 2112, each PROM containing a full page of memory, and need not be connected in pairs. Word selection within each PROM is made by the low-order memory bus--DL7 through DL0. Page selection is accomplished through each PROM's chip enable ($\overline{CE}$) terminal. There are eight chip-enable lines associated with the 403 PROMs--$\overline{CE0}$ through $\overline{CE7}$--corresponding to pages 000 through 007 in memory.

Like the 2112, the 1702A has three-state bus driver data terminals. However, in the case of the PROM these output terminals are unidirectional only, since the CPU cannot normally write into these devices. The output terminals leave the high-impedance third state when the PROM is chip-enabled, driving the 403's internal data bus, BB7 through BB0.

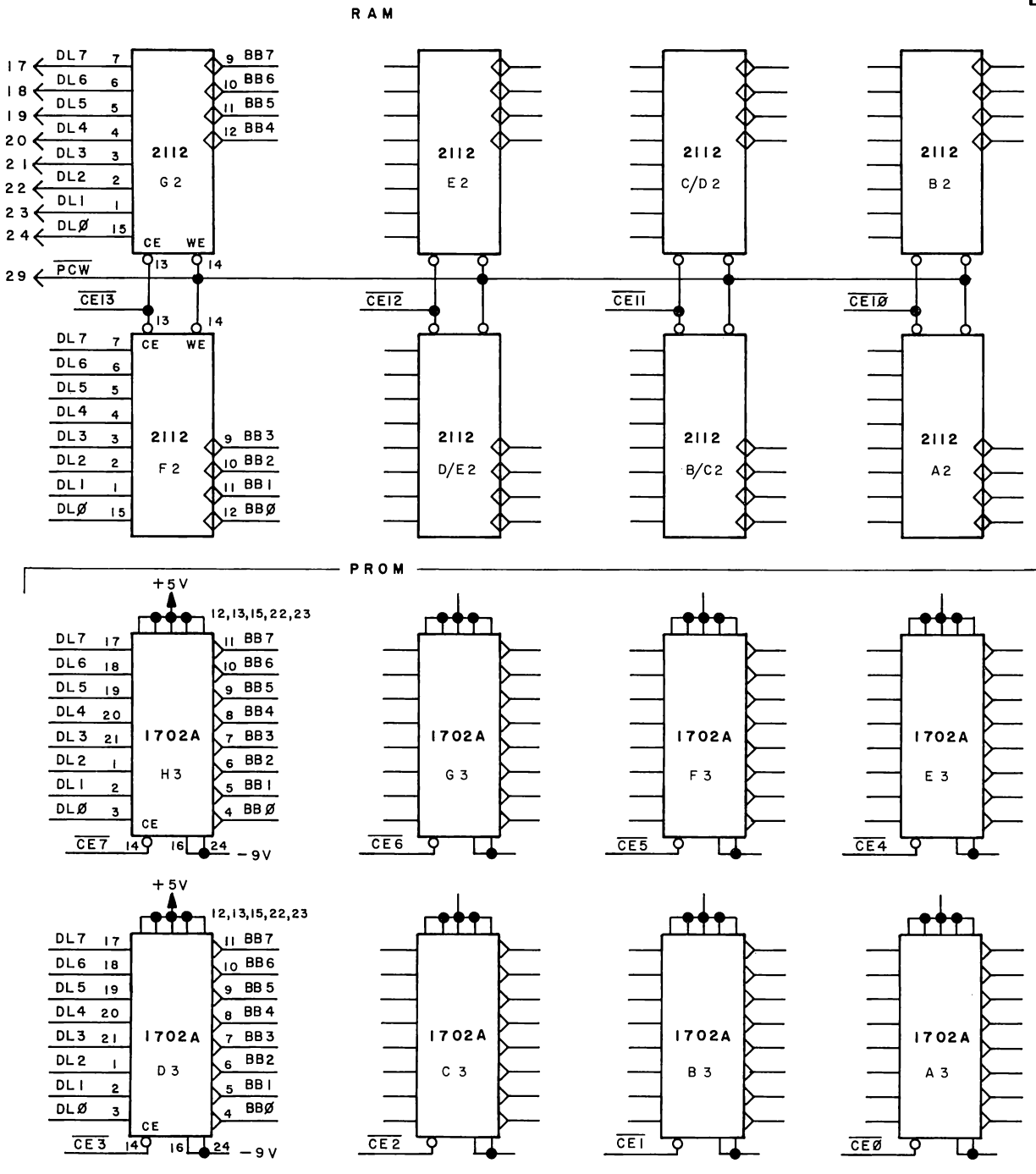
The 1702A PROM is programmed electrically at the factory with specialized programming equipment. The device retains programmed data indefinitely under normal conditions. Equipped with a quartz window on the top--through which the user can see the silicon chip on which the memory circuitry is etched--the 1702A can be erased by exposure to a strong source of ultra-violet light.

In many microcomputers, the first PROM, at page 000, has special significance in the system. This is because the computer software includes RESTART instructions which cause the CPU to jump to one of a number of designated locations in page 000. This is the location for the single-chip Monitor programs supplied with the *modular micro* series.

1.3 *Memory Decoding* The 403 PROM/RAM board includes memory decoding circuitry which generates the twelve chip enable signals which select the four pages in RAM, and eight pages in PROM. The CPU uses the high-order address bus, DH7 through DH0, to address pages in memory (just as the DL bus selects words within those pages).

©1975, Martin Research. Pp. 1-10, Rev. A, Nov. 1975.

REV
A



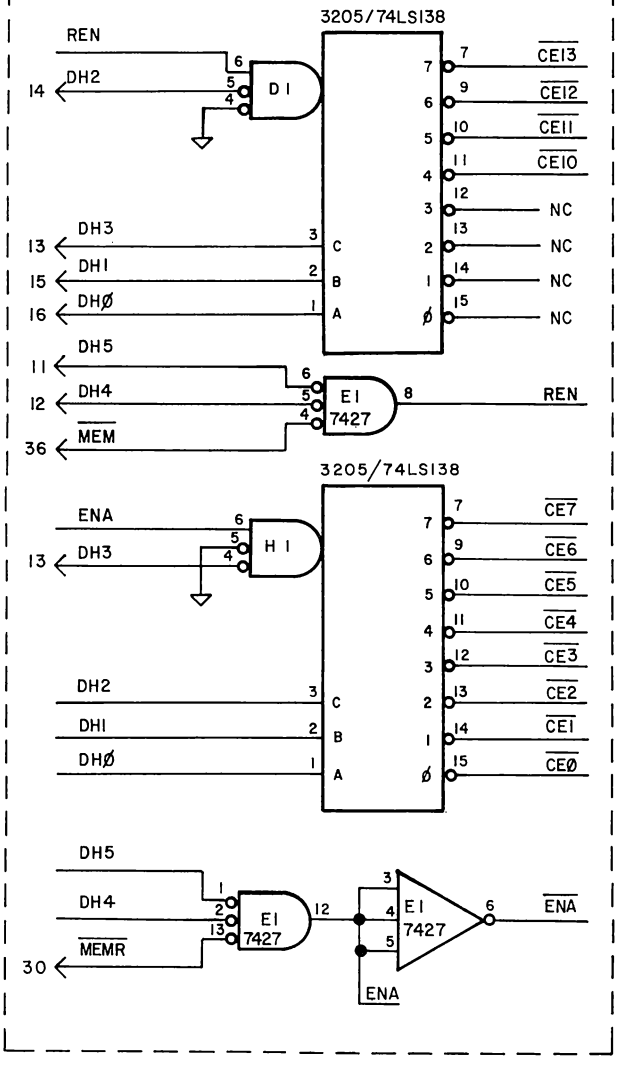
NOTE: PROPRIETARY INFORMATION. NOT TO BE DISCLOSED WITHOUT EXPRESS WRITTEN PERMISSION.	DRAWN BY Schneider	CHECKED BY DK	DRAWING NO. DS 403 (Page 3)
	DATE 11-19-75	DATE 11-19-75	

EF4A

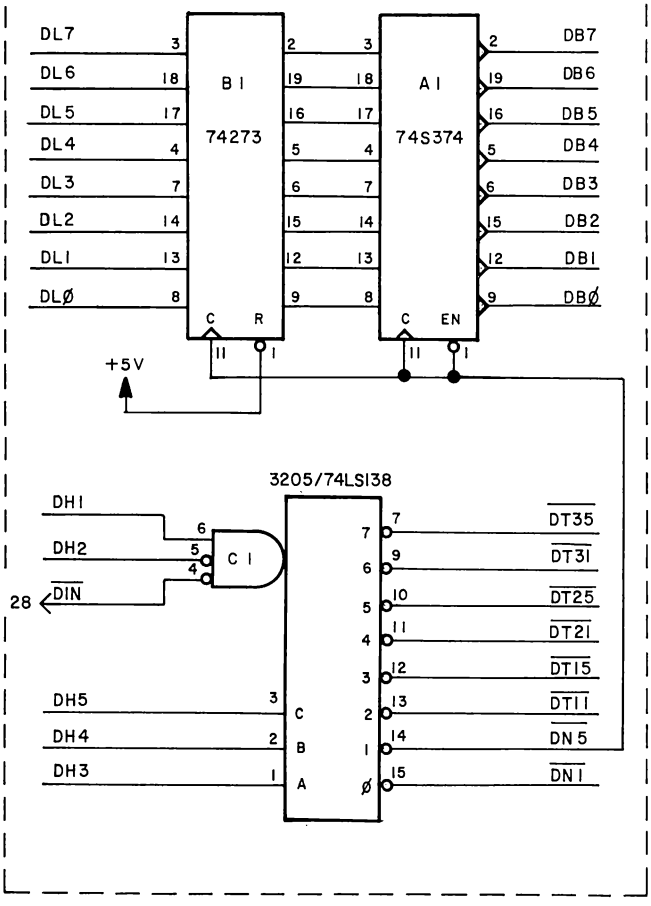
modular micros martin research

REV
A

MEMORY DECODERS



OPTIONAL SHORT STACK



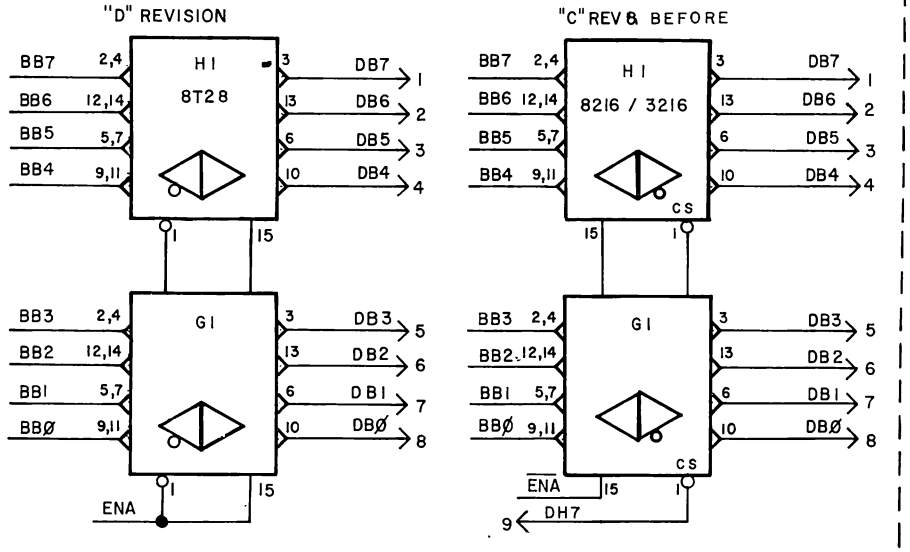
NOTE: PROPRIETARY INFORMATION. NOT TO BE DISCLOSED WITHOUT EXPRESS WRITTEN PERMISSION.	DRAWN BY Schneider	CHECKED BY DM	DRAWING NO. DS 403 (Page 4)
	DATE 11-19-75	DATE 11-19-75	

EF4A

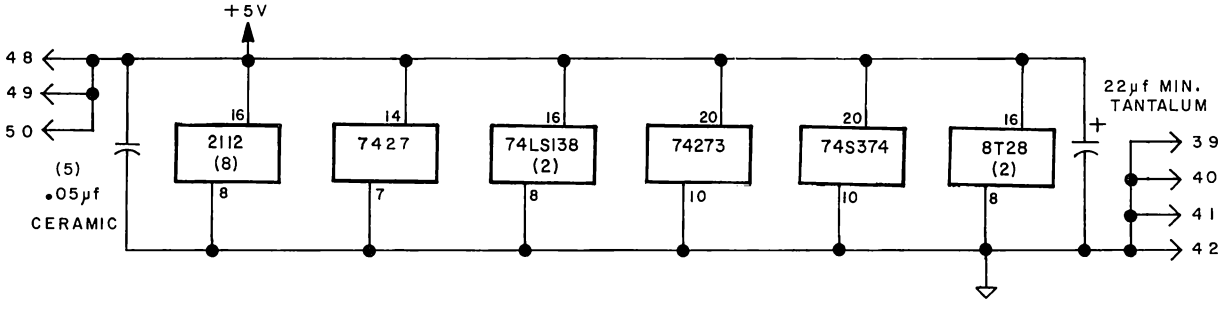
modular micros
martin research

REV
A

BUS DRIVERS



POWER BUSSING



NOTE: PROPRIETARY INFORMATION. NOT TO BE DISCLOSED WITHOUT EXPRESS WRITTEN PERMISSION.	DRAWN BY Schneider	CHECKED BY DK	DRAWING NO. DS 403 (Page 5)
	DATE 11-19-75	DATE 11-19-75	

EF4A

modular micros martin research

SEC. 1 CIRCUIT DESCRIPTION (cont'd)

There are four pages in a *K* of memory--that is, 1,024 bytes, or a *kilobyte.* There are 8K in a *bank,* and in a computer using all eight of the DH bus lines, a total of eight banks--or 64K words of memory. Whether all eight DH lines are actually used depends on the amount of memory in the system, and also on the type of CPU in the computer.

The PROMs on the 403 board reside in the first eight pages of memory, or the first 2K. The RAM resides in the next four pages, or the third K. Fig. 1.3.1 illustrates the relation between the DH register bits and the bank, K, and page address of the 403's memory elements.

BANK			K			PAGE					
DH7	DH6	DH5	DH4	DH3	DH2	DH1	DH0	BANK	K	PAGE	CHIP
0	0	0	0	0	0	0	0	0	0	0	PROM 0
0	0	0	0	0	0	0	1	0	0	1	PROM 1
0	0	0	0	0	0	1	0	0	0	2	PROM 2
0	0	0	0	0	0	1	1	0	0	3	PROM 3
0	0	0	0	0	1	0	0	0	1	0	PROM 4
0	0	0	0	0	1	0	1	0	1	1	PROM 5
0	0	0	0	0	1	1	0	0	1	2	PROM 6
0	0	0	0	0	1	1	1	0	1	3	PROM 7
0	0	0	0	1	0	0	0	0	2	0	RAM 10
0	0	0	0	1	0	0	1	0	2	1	RAM 11
0	0	0	0	1	0	1	0	0	2	2	RAM 12
0	0	0	0	1	0	1	1	0	2	3	RAM 13

Figure 1.3.1--DH Register Addressing of 403 Memory Pages

Figure 1.3.2 shows how the DH and DL address bits correspond with split octal notation, frequently used in eight-bit microcomputers with eight high-order and eight low-order address lines (DH and DL busses). The address shown is location 013 370, a word in page 013 in RAM on the 403 board, near the end of that page.

ADDRESS BITS	DH7	DH6	DH5	DH4	DH3	DH2	DH1	DH0	DL7	DL6	DL5	DL4	DL3	DL2	DL1	DL0
BINARY	0	0	0	0	1	0	1	1	1	1	1	1	1	0	0	0
SPLIT OCTAL	0		1			3			3		7			0		

Figure 1.3.2--DH Register Bits and Split Octal Notation

The 403 board employs two 74LS138 strobe decoders and associated gates to decode the twelve chip enable signals required. The upper 7427 three-input NOR gate (drawn as a negative

SEC. 1 CIRCUIT DESCRIPTION (cont'd)

AND) produces an enable signal, *REN,* which is high only when DH5, DH4, and $\overline{\text{MEM}}$ are low. $\overline{\text{MEM}}$, a signal created by the CPU board, is a composite signal indicating that the CPU data bus is ready *either* to read from memory, or write into memory. REN and DH3 through DH0 are the inputs to a 74LS138/3205 decoder (the upper decoder on the 403 schematic), which produces the required chip enable strobes for the RAM chips. The signal $\overline{\text{PCW}}$, produced by the CPU board and indicating that the CPU is in a memory write mode, is connected to the write enable terminals of the 2112s, and controls whether a selected pair of 2112s is in read or write mode.

The lower pair of 7427 gates produce the signal *ENA* (and its complement $\overline{\text{ENA}}$), a signal which is similar to REN but is valid only when $\overline{\text{MEMR}}$ is low--that is, during memory read cycles. The difference is explained by the fact that the PROMs should not be enabled during memory write cycles. The DH register inputs are similar to those of the other decoder, with the DH2 and DH3 bits swapped. Study of Fig. 1.3.1 demonstrates that this arrangement produces the eight required PROM strobes.

NOTICE--The 403 memory decoders do not decode the DH7 and DH6 bits. See Sec. 1.5 for restrictions on memory use which result.

1.4 *Bus Drivers* In order to permit the 2112s and 1702As on the 403 board to drive the *modular micro* bus, a bus driver stage is required. Two four-bit bidirectional bus drivers connect the PROM/RAM board's internal data bus, BB7 through BB0, with the *modular micro* data bus, DB7 through DB0.

In the *D* revision of the 403, two 8T28 devices are used. When ENA is high, DH5 and DH4 are low (indicating selection of the 403 board) and the CPU is in the memory read cycle. Pin 15 is at the high level, and the bus-driving outputs shown at the right of the package (in the 403 schematic) are activated, causing the data on the internal data bus to be passed onto the *modular micro* data bus. The high level at pin 1 disables the outputs at the left side of the package, allowing these terminals to be used as inputs. During memory write cycles, the drive direction is reversed, and the 2112 RAMs may be selected to receive data. Actually the 403 internal data bus receives information from the DB bus at all times, even when it is not really meant for the 403 board (except during memory read cycles referencing the 403). This is permissible because only at the appropriate times are the 2112s enabled to receive information from the internal data bus.

In the *C* and earlier 403 versions, two 8216/3216s are used. Here pin 15 is a direction control. When $\overline{\text{ENA}}$ is low, the bus drivers point outwards, driving memory data from the 403 to the CPU. During memory write cycles, the bus drivers point inwards, transferring data from the DB bus to the internal data bus. Again, the internal data bus sometimes receives data not meant for the 403 board, but the 2112s listen only when selected. Note that the 8216 chip select (CS) terminal is connected to the DH7 address line. Only when DH7 is low can the bus driver transfer data; when DH7 is high, all 8216 data terminals are in a high-impedance state. The results are explained in the following section.

SEC. 1 CIRCUIT DESCRIPTION (cont'd)

1.5 *Memory Addressing Restrictions* The 403's memory decoder stage does not gate in the DH7 and DH6 bits. In a *MIKE 2* system, based on the 8008, this is unimportant because the CPU does not use these bits for memory addressing. (The DH7 and DH6 lines are grounded on the 8008 CPU board.) But the 403 places certain restrictions on a system in which all sixteen address bits are to be used.

The 403 bus drivers will drive the *modular micro* data bus not only when the CPU has selected the board's nominal addresses in memory--in bank 0, in the first 3K--but also at certain *ghost addresses.* The most important case occurs when the DH5 through DH0 bits in the ghost address are the same as those in the 403 address--but the DH7 and DH7 bits are not. This limitation does not apply unless the user needs more than 16K of memory.

Another factor is that the 403 bus drivers drive the DB bus during a DH3/DH2 address combination (1/1) which is not used by chips on the board--that is, when the CPU tries to reference pages 014 through 017.

The ghost page numbers to be avoided with the 403 are 014 through 017, 100 through 117, 200 through 217, and 300 through 317.

In the *C* and earlier versions of the 403, the DH7 bit must be low to enable the bus driver, as explained earlier. In effect these boards decode all but the DH6 bit. As a result, ghost pages are reduced to the following: 014 through 017; 200 through 217.

The Model 423 PROM/RAM board, an improved version of the 403, avoids these restrictions.

1.6 *Memory Speed* Two factors determine the effective speed at which the 403 PROM/RAM board can run: the speeds of the PROM and RAM chips used; and the mode of memory addressing employed by the CPU in the microcomputer.

The 1702A-6 PROM and 2112 RAM may be characterized in two ways: by their *access time,* and by their *select time. Access time* measures the period starting when a valid word address is presented to the memory, and ending when valid data is guaranteed to have been read from or written into memory.

When using the *select time* spec, it is assumed that a valid word address is already available, and the critical period begins when the memory is activated at its chip select (or chip enable) terminal, ending when valid data transfer is guaranteed. Normally the select time is appreciably shorter than the access time.

(The time it takes for memory decoding circuitry--like the 74LS138--to transfer address signals is also a factor. But this bipolar circuitry is so much faster than MOS memory that these delays are comparatively very small.)

SEC. 1 CIRCUIT DESCRIPTION (cont'd)

In a *MIKE 2* microcomputer, the 8008 CPU sends out a DL address first, giving plenty of time for a given word to be selected within a 2112 or 1702A. Then the DH address bits become valid, activating the 403 memory decoder stage, which selects the proper chip through its chip select terminal. Finally, a data transfer occurs between memory and CPU. The period which determines effective memory speed is *select time.*

In a *MIKE 3* microcomputer, the 8 080 CPU provides considerably less gap between the time when an address becomes valid, and the time when a memory read/write may occur. As a result, the effective speed is determined by the memory component's *access time.* In practice, the CPU may have to enter a WAIT state for a slow memory component. (Further details: see the data sheet for the 8080 CPU board.)

The 2112 RAM has an access time of 1.0 us and a chip select time of 800 ns. The 1702A-6 PROM has an access time of 1.5 us and a chip select time of 900 ns. These chip select times are adequate for full-speed operation in a *MIKE 2* microcomputer operating with an 8008-1 CPU.

1.7 *Short Stack Option* Space is provided on the 403 board for three optional ICs, making up the *Short Stack* option. Because of availability problems with the 74S374, room for this option has been made on the Model 441 CPU board, using alternate ICs. Or it can be assembled on a *modular micro* breadboard. For details, see the data sheet on the *Short Stack,* KT9000.

SEC. 2 PARTS LIST

2.1 *Standard Parts* The following parts are supplied with the 403 with 256 bytes of PROM and 256 bytes of RAM. This is the version of the 403 supplied with the *MIKE 2* microcomputer.

Part No.	Quantity	Description
BD403	1	PC BOARD, PROM/RAM
1000	1	CONNECTOR, PC board header, 50 pin, rt. angle solder leads
1012	1	SOCKET, 14-pin DIP, solder-tail
1013	6	SOCKET, 16-pin DIP, solder-tail
1037	1	SOCKET, 24-pin DIP, solder-tail, gold
2016	1	IC, 7427 8-input NAND
2042	2	IC, 74LS138 3-to-8 decoder
2130	2	IC, 8T28B quad bi-bus driver (early version used P/N 2133, 8216: see Sec. 1.4 above)
2200-13	1	IC, 1702A-6 256 x 8 PROM, with *MIKE 2* Monitor
2205	2	IC, 2112 256 x 4 RAM, 1.0 us access time
5501	3	CAPACITOR, .05 to 0.1 uf, 10 V min., ceramic
5502	1	CAPACITOR, 22 to 50 uf, 15 V min., tantalum

SEC. 2 PARTS LIST (cont'd)

2.2 *Memory Expansion* The following parts may be added by the user to fill out the memory capacity of the 403 PROM/RAM board. Note that preprogrammed PROMs are available; see our current catalog and Software Package data sheets. For prices on parts, see our Replacement Parts List.

Part No.	Quantity	Description
1012	---	SOCKET, 16-pin DIP, solder-tail (for RAMs)
1037	---	SOCKET, 24-pin DIP, solder-tail, gold (for PROMs)
2200	---	IC, 1702A-6 256 x 8 PROM (unprog.)
2205	---	IC, 2112 256 x 4 RAM, 1.0 us access time

SEC. 3 ASSEMBLY NOTES

3.1 *RAM and PROM Locations* Figure 3.1.1 illustrates the locations of the RAM and PROM chips on the 403 board. The 403-2 is assembled with the *MIKE 2* Monitor PROM at page 000, and two 2112s at left center, making up page 010. The component side of the board is shown. Page 013 in RAM should be installed for full CPU interrupt handling, as described in the data sheet on the central processing unit.

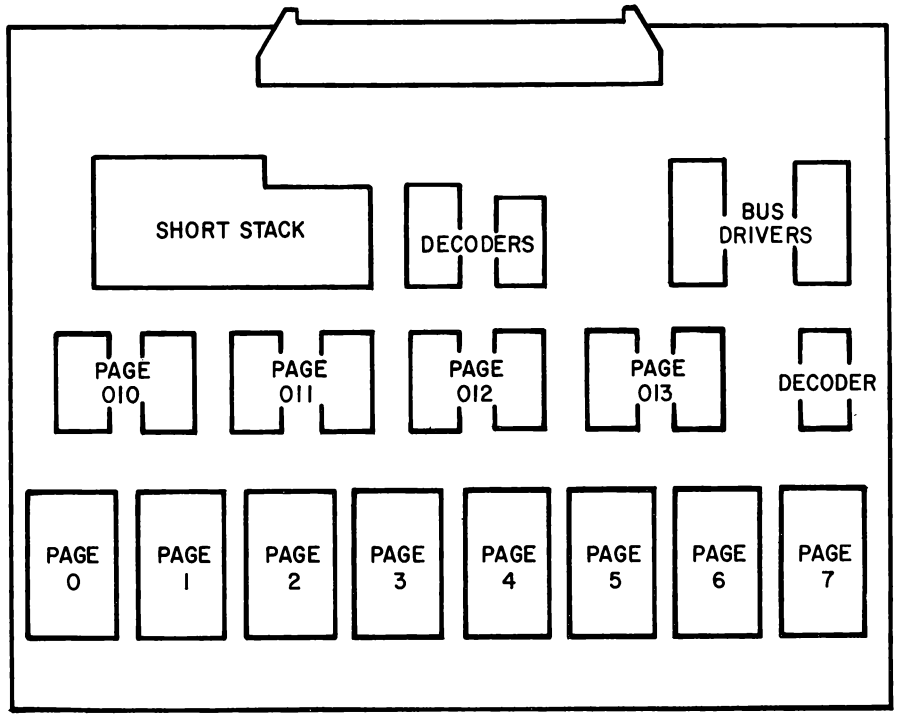


Figure 3.1.1--Page Addresses of RAM and PROM on 403 Board

The *FOUR KILOBYTE RANDOM ACCESS MEMORY* is a member of the family of modular microcomputers from Martin Research. Consisting of 4,096 eight-bit bytes of memory storage (for a total of 32,768 bits), this 4K RAM is fully compatible either with an 8080 system--the *MIKE 3*--or with an 8008 system--the *MIKE 2.* Featuring static memory devices, the 405 includes full memory address decoding, with user-programmable addresses. An optional WAIT circuit on the board allows the user to accomodate slow RAM chips to a fast CPU. The board requires a single +5 volt supply, and uses standard 50-pin connectors for plug-in interfacing with the *MIKE 3/MIKE 2* bus.

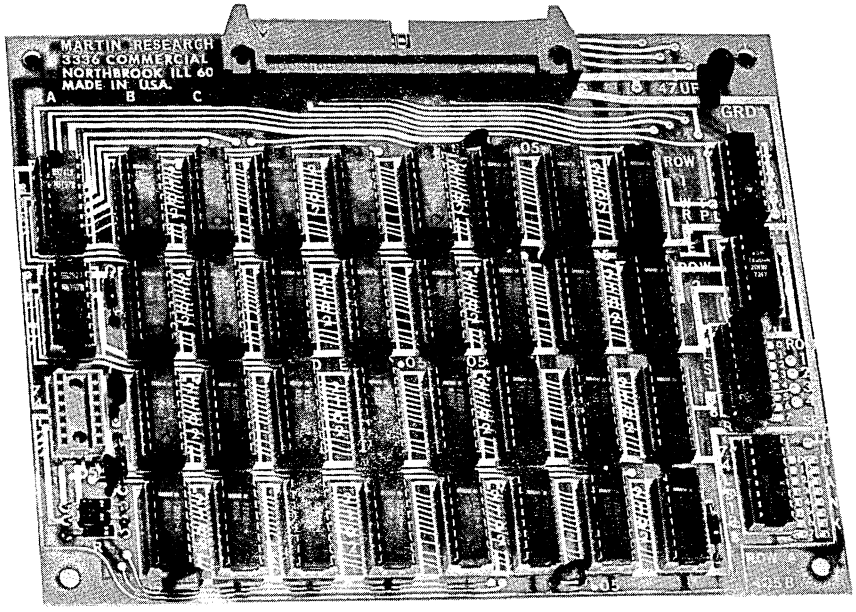
SPECIFICATIONS

Speed — 450 nanoseconds access time is standard for 4K RAMs purchased from Martin Research. This speed is suitable both for our 8080 system (*MIKE 3)* and for the 8008 *(MIKE 2).*

Power — Requires +5 volt supply, regulated ±5%. Typical current requirement for full complement of 4K, using 32 type 2102A-4 RAM chips, 1.0A.

Interfacing — Connects directly to the modular micro bus with fifty-pin connectors. Memory address is user-selectable using jumpers on the board.

Mechanical — Standard size for Martin modular micro family: 5.5 by 7.0 inches (140 by 178 mm). Printed circuit boards are manufactured by commercial PC houses to meet professional standards, with plated-through holes, solder-mask protection of traces on the solder side of the board, and silkscreened part outlines for ease in assembly and repair.



REV. A

TECHNICAL DATA

SEC. 1 CIRCUIT DESCRIPTION

1.1 *The 2102 RAM* The 2102 static RAM is the basic building block of the 4K memory board. Each 2102 contains a matrix of 1,024 memory cells. Referring to the diagram of the 2102 in the *Typical Circuit* section of the 405 schematic, the 2102 has ten address lines (*A9-A0*), a chip enable ($\overline{CE}$) and a write enable ($\overline{WE}$) control, and separate data input (*D/IN*) and data output (*D/OUT*) terminals.

The ten address lines are necessary to select one of the 1,024 internal memory cells. ($2^{10} = 1{,}024$.) A binary *1* or *0* may be written into a selected cell via the chip's *D/IN* terminal, and read out via its *D/OUT* terminal. The 2102 is disabled when the chip enable ($\overline{CE}$) terminal is high, and cannot be read from or written into. Stored data remains stable indefinitely, however, so long as power is supplied.

The $\overline{WE}$ terminal controls whether the 2102 is in the READ or WRITE mode. When this terminal is low, information on the *D/IN* pin is written into the selected memory cell--provided CE is low. When the $\overline{WE}$ pin is high, the 2102 is in the READ mode (again providing that the chip is enabled); information stored in the addressed memory cell appears at the D/OUT terminal.

The 2101's output voltages are TTL-compatible. Since the 2102 incorporates three-state circuitry at its data output terminal, it goes into a high-impedance state when the chip is not in the READ mode, allowing some other device on the same data bus to determine the voltage level.

These features allow one to build up a relatively large memory block with a minimum of extra circuitry.

1.2 *Memory Structure* The 2102 may be thought of as having 1,024 (or 1K) one-bit words. Eight are required to interface with an eight-bit microcomputer. The minimal amount of memory on the Model 405 is thus eight 2102s, making up a 1K x 8 array. Fully stuffed, the 405 uses 32 2102 RAM chips, for a 4K x 8 array.

The Model 405 allows the user to select the locations in the microcomputer's memory at which the board's four kilobytes are addressed. One wire jumper on the board places the 405 in any one of the eight *banks* which are possible in an eight-bit microcomputer using a sixteen-bit address bus. Each bank is made up of 8K words of memory. Four other wire jumpers are used to assign positions within the selected bank to each of the four *KILs (kilobytes,* i.e., 1,024 words) of memory on the 405 board. In turn, each of these KILs corresponds to a row of eight 2102s on the 405 board.

Memory address bits, bank/KIL/page addresses, and octal page numbers are shown in Figure 1.2.1. Several commonly-used memory locations for the 405 are illustrated. The chart is divided into groups of two lines, where the first line shows the address of the first page in a given 4K block, and the second line shows the address of the last page in the same 4K block.

©1975, Martin Research. Pp. 1-9, Rev. A, Nov. 1975.

SEC. 1 CIRCUIT DESCRIPTION (cont'd)

BANK			KIL			PAGE					OCTAL	
DH7	DH6	DH5	DH4	DH3	DH2	DH1	DH0	BANK	KIL	PAGE	PAGE	COMMENTS
0	0	0	1	0	0	0	0	0	4	0	020	
0	0	0	1	1	1	1	1	0	7	3	037	
0	0	1	0	0	0	0	0	1	0	0	040	Location of Factory
0	0	1	0	1	1	1	1	1	3	3	057	Assembled 405
0	0	1	1	0	0	0	0	1	4	0	060	Last 4K Block
0	0	1	1	1	1	1	1	1	7	3	077	in 8008 System
0	1	0	0	0	0	0	0	2	0	0	100	First 4K Block
0	1	0	0	1	1	1	1	2	3	3	117	in Full 64K
1	1	1	1	0	0	0	0	7	4	0	360	Last 4K Block
1	1	1	1	1	1	1	1	7	7	3	377	in Full 64K

Figure 1.2.1--Typical Locations of 4K RAM in Modular Micro System

As noted on the chart, factory-assembled 405 boards are addressed at octal pages 040 through 057. Of course the user can change the addresses. The *MIKE 2* user should recall that DH7 and DH6 are always zero in an 8008 system, meaning that page 077 is the last page in the 16K memory capacity of the *MIKE 2.* And, finally, the last two 4K blocks in a full 64K memory array--pages 300 through 377, defined by the condition that DH7 and DH6 are high--may be allocated for 8008-compatible I/O ports in *modular micro* systems based on the 8080 or other processors. (For details see the appropriate CPU data sheet.)

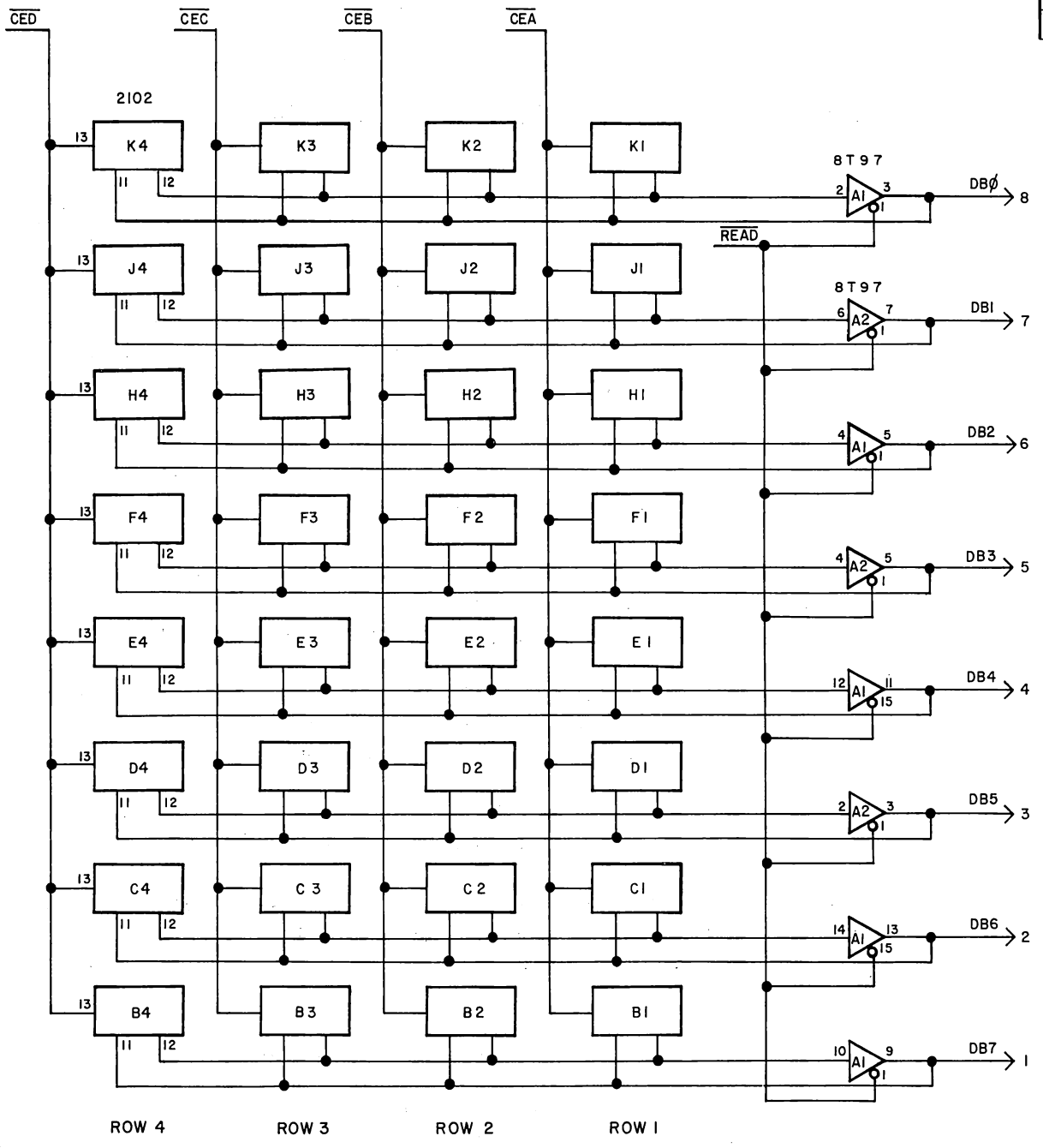
1.3 *Memory Decoding* A 74LS138/3205 decodes the three high-order microcomputer address lines, DH7, DH6, and DH5. The eight possible combinations of the states of these bits are shown on the inside of the decoder on the 405 schematic. The one *BANK* signal which is low at any given time indicates which bank in memory is selected by the address bus. A user-selectable wire jumper (or *strap*) is connected to the following decoder, determining which of the eight bank addresses will select the memory on the 405 board.

The following 'LS138 decodes the DH4, DH3, and DH2 address lines, selecting a *K (KIL)* address within the selected bank. The active-low enable terminals of this stage must be activated by the bank-selector to produce an output. There are four active-low outputs, labeled $\overline{\text{KIL7}}$ through $\overline{\text{KIL0}}$, corresponding to the eight kilobytes within a bank. User-selectable straps program the 405's four kilobytes with *K* addresses within the selected bank.

The four resulting chip-enable signals are labeled $\overline{CEA}$ through $\overline{CED}$. $\overline{\text{CEA}}$ enables the eight 2102s on ROW 1, $\overline{\text{CEB}}$ enables ROW 2, etc., as shown on the 405 schematic.

FOUR KILOBYTE RANDOM ACCESS MEMORY
MODEL 405

REV
A



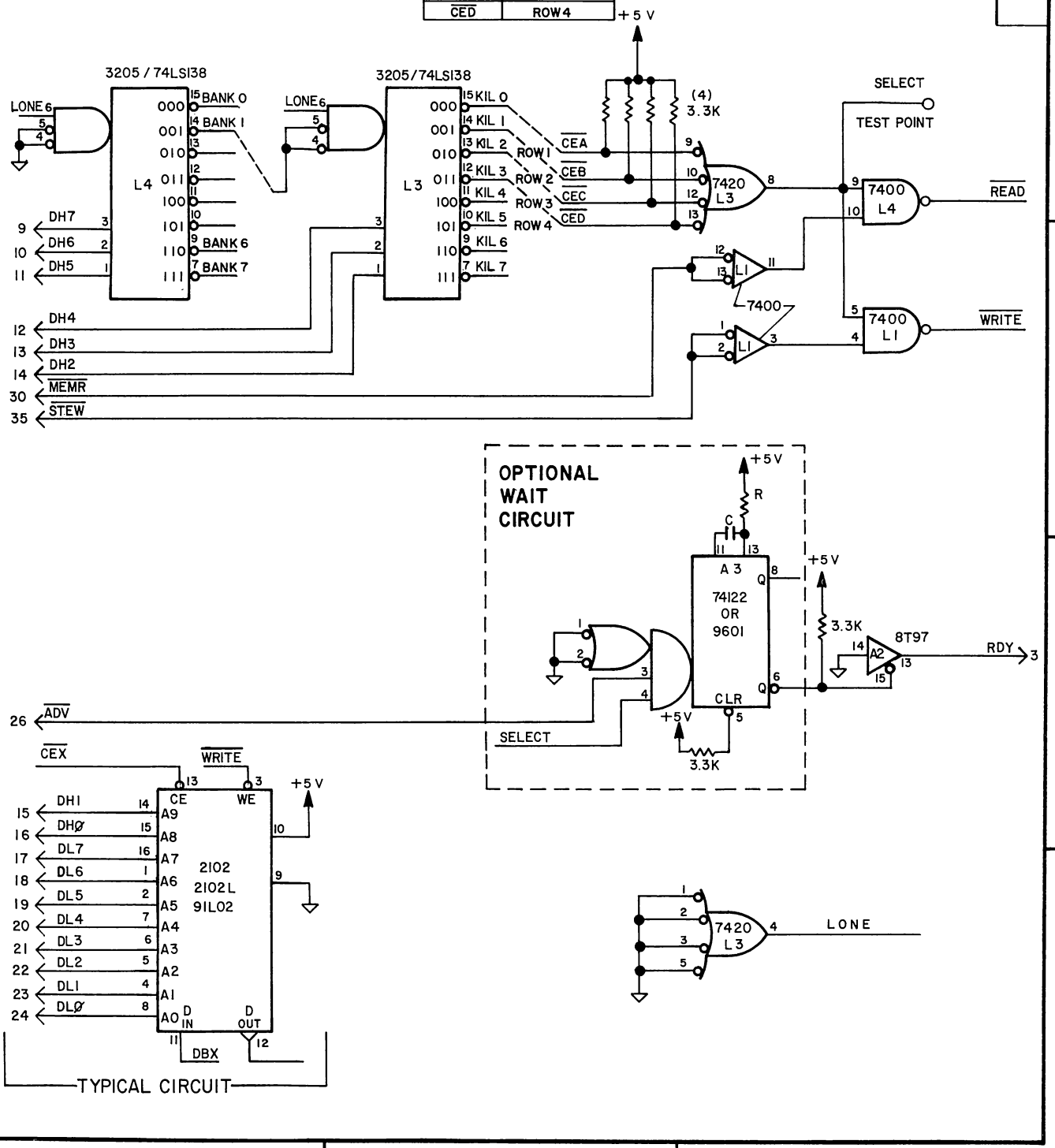
NOTE: PROPRIETARY INFORMATION. NOT TO BE DISCLOSED WITHOUT EXPRESS WRITTEN PERMISSION.	DRAWN BY ROECK DATE 11-10-75	CHECKED BY DATE
		DRAWING NO. DS405 (Page 4)

EF4A

modular micros
martin research

REV
A

SIGNAL	ENABLES
$\overline{CEA}$	ROW 1
$\overline{CEB}$	ROW2
$\overline{CEC}$	ROW3
$\overline{CED}$	ROW4



NOTE: PROPRIETARY INFORMATION. NOT TO BE DISCLOSED WITHOUT EXPRESS WRITTEN PERMISSION.

DRAWN BY SCHNEIDER	CHECKED BY DK
DATE 11-10-75	DATE 12/1/75

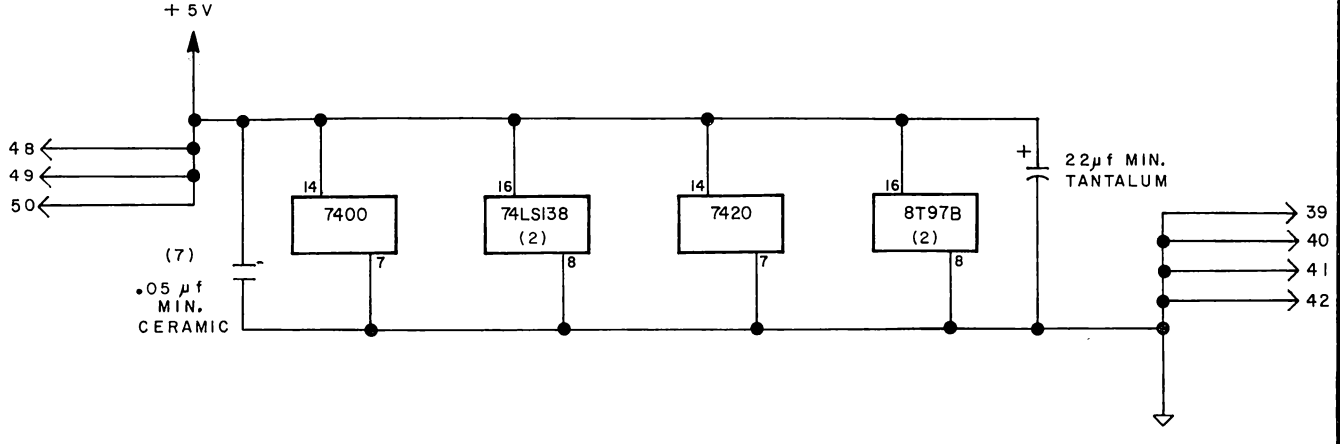
DRAWING NO. DS 405 (Page 5)

EF4A

modular micros
martin research

REV
A

POWER BUSSING



NOTE: PROPRIETARY INFORMATION. NOT TO BE DISCLOSED WITHOUT EXPRESS WRITTEN PERMISSION.	DRAWN BY ROECK / DATE 11-10-75	CHECKED BY D.R.C / DATE
		DRAWING NO. DS 405 (Page 6)

EF4A

modular micros martin research

SEC. 1 CIRCUIT DESCRIPTION (cont'd)

When any one of the blocks of memory on the 405 board has been selected, a 7420 4-input NAND (functioning here as a negative OR) produces the *SELECT* signal. Available as a test point, marked *SEL TP* at location L3 on the board, the SELECT line can be used to verify the operation of the memory decoding circuitry. Note however that this line may include short positive pulses (glitches) even when a 405 board address has not been selected. One DH address bit may change slightly more rapidly than another, producing a false address momentarily. This does not affect 405 board operation, since SELECT is properly synchronized with the CPU before being used to enable RAM.

Two 7400 NAND gates are used to gate SELECT with CPU-generated signals which control memory read and write cycles. *$\overline{MEMR}$* is used to produce the signal *$\overline{READ}$,* and $\overline{STEW}$ (sometimes called $\overline{MEMW}$) produces *$\overline{WRITE}$.* $\overline{WRITE}$ is connected to each 2102 and determines whether the RAM is in a READ or WRITE mode. $\overline{READ}$ controls the board's bus drivers (described below). For test purposes, these signals are available: *R TP* ($\overline{READ}$) at location L1, and *WRITE TP* at location K1.

Unused terminals of the decoders are connected to *LONE* (logic one), a signal close to +5 V. On the 405, LONE is generated by a 7420, functioning as an inverter with grounded inputs.

1.4 *Bus Drivers* The 405 board includes 8T97 three-state bus drivers to amplify the data output power of the 2102s, enabling them to drive an expanded *modular micro* system. The data output terminals of the 2102s are connected to the bus driver inputs. Effectively there is an *output bus* on the 405 board, with four 2102s (corresponding to the four rows of chips) on each of eight lines. With their output terminals normally in an inactive, high-impedance state, the bus drivers transmit the data on the output bus to the CPU data bus (DB7-DB0) when they are activated by the $\overline{READ}$ signal. This occurs only when an address on the 405 board is to be read by the CPU.

The 8T97 outputs can *source* up to 5.2 MA (at logical one TTL voltage levels) and *sink* up to 48 MA (at logic zero TTL voltage levels). This amounts to a drive capability of 30 standard TTL loads.

The data input terminals of the 2102 RAM chips are connected directly to the microcomputer's data bus, DB7-DB0. Load current for each 2102 data input terminal amounts to only 10 uA maximum, or 40 uA maximum per data bus line--a load which can easily be driven in even a small microcomputer.

The 2102 address terminals, D9 through D0, also require a maximum of 10 uA of input current. A fully-stuffed 405 board, with 32 chips, imposes a theoretical maximum of 320 uA on the DH1, DH0, and DL7-DL0 address bus lines. In practice the load is much less.

SEC. 1 CIRCUIT DESCRIPTION (cont'd)

1.5 *Memory Speed* 450 ns access time is standard for RAM chips supplied by Martin Research for use on the Model 405 4K RAM board. The access time of the 2102A-4 RAM provided matches the speed of the central processing unit in most *modular micro* systems, avoiding the need to make the CPU wait for memory access. For example, it allows the 8080 in the standard *MIKE 3* system to operate without wait cycles. The user is cautioned against the use of slower RAM, to avoid compatibility problems if the system is upgraded at a later date. Slow memory can be the weak link in a supposedly fast microcomputer.

Still faster 2102 chips (or 2102 equivalents) can be supplied as special options.

1.6 *Optional WAIT Circuit* When the memory components on the 405 board are insufficiently fast to meet the requirements of the CPU, the user must make the CPU *wait* for memory access. In the *modular micro* series, the selected slow memory component must pull down the *READY* line on the microcomputer bus. Also called $\overline{BSY}$, indicating that the system is *busy* when the line is low, this condition causes the CPU to enter an idle state while the memory components complete internal decoding. When the line goes high again, the memory cycle is completed and the CPU continues.

When a 74122 or 9601 one-shot multivibrator is installed at location A3, it is triggered each time that the 405 board is selected. The SELECT line is gated with the $\overline{ADV}$ (address valid) signal, causing the one-shot to fire as this signal goes high (indicating the address lines are stable). The $\overline{Q}$ output goes low for a period of time determined by the values of *R* and *C* at pins 11 and 13. This activates an 8T97 three-state bus driver stage, causing its output to leave the high-impedance state. Since the driver's input is grounded, it pulls the $\overline{BSY}$ line down to ground.

The WAIT circuit is an option on other *modular micro* components as well as the 405 board. Where best to install the circuit (if it is necessary) depends on system memory and CPU speed considerations. Further details are provided in the data sheets on the 8080 CPU board and the PROM/RAM board.

The 405 optional WAIT circuit is disabled simply by removing the 74122 or 9601. A pull-up resistor ensures that the 8T97 driver will not be enabled when the one-shot has been removed.

SEC. 2 PARTS LIST

2.1 *4K RAM* The following parts are supplied with a 405-4, a fully-stuffed Model 405 with 4K of 450 ns RAM. Partially-stuffed versions are also available, as described in current *modular micro* price lists.

Part No.	Quantity	Description
BD405	1	PC BOARD, 4K RAM
1000	1	CONNECTOR, PC board header, 50-pin, rt. angle solder leads
1012	3	SOCKET, 14-pin DIP, solder-tail, tin

REV. A

SEC. 2 PARTS LIST (cont'd)

Part No.	Quantity	Description
1013	36	SOCKET, 16-pin DIP, solder-tail, tin
2000	1	IC, 7400 quad NAND gate
2014	1	IC, 7420 dual 4-input NAND
2042	2	IC, 74LS138 3-to-8 decoder
2132	2	IC, 8T97B hex bus driver
2207	32	IC, 2102A-4 1024 x 1 RAM, 450 ns
4000-55	6	RESISTOR, 3.3K, 10%, ¼w
4000-60*	1	RESISTOR, 8.2K, 10%, ¼w
5000-16*	1	CAPACITOR, 220 pf, 10%, 1000 V, ceramic
5501	7	CAPACITOR, .05 to 0.1 uf, 10 V min., ceramic
5502	1	CAPACITOR, 22 to 50 uf, 15 V min., tantalum

2.2 *WAIT Option* The following part may be added to the 405 board to supply the optional WAIT function (Sec. 1.6 above). The necessary IC socket is provided with the basic 405 board, as are an 8.2K resistor and a 200 pf capacitor (indicated with asterisks in Sec. 2.1 above). These values of *R* and *C* are for a memory access time of 1.0 us and a CPU cycle time of 500 ns. The WAIT circuit is activated simply by installing the one-shot, and disabled by removing it.

Part No.	Quantity	Description
2037	1	IC, 74122 one-shot

The *423 PROM/RAM BOARD* is a basic memory component for computer systems in the *modular micro* family. The 423 has room both for 2K of programmable read-only memory (PROM), and for 1K of random access memory (RAM). Typically the PROM is used for storage of programs essential to system operation--such as the *MIKE 3* or *MIKE 2* Monitors. The RAM is for storing user's programs, for temporary data storage during program execution, and for storing CPU register status during subroutine and interrupt handling.

An optional WAIT circuit on the board allows the user to accommodate slow memory chips to a fast CPU. In addition, the memory address of each block of PROM and RAM is user-selectable. Both architecture and control structure of the 423 make it suitable for interfacing with a variety of *modular micro* systems.

SPECIFICATIONS

Memory Capacity	Up to 2K bytes of PROM: uses up to eight type 1702A, 256 by 8 reprogrammable memory devices. Up to 1K of RAM: up to eight type 2112 256 by 4 static RAM chips.
Speed	RAM, 800 ns select time; PROM, 900 ns select time (meets requirements of 8008-1 system without memory wait cycles). Faster memory chips available as options.
Interfacing	Connects directly to the *modular micro* bus with standard fifty-pin connectors. On-board bus drivers ensure adequate driving power for expanded systems. Typical page addresses: PROMs, pages 000 through 007; RAMs, pages 010 through 013. However, memory addresses are user-selectable using jumpers on the board.
Power	With 256 bytes each of PROM and RAM, requires +5 V, 380 MA; -9 V, 35 MA; regulated ±5%. (-9 V can be derived from a -12 V supply using optional on-board diodes.)

REV. A

TECHNICAL DATA

SEC. 1 CIRCUIT DESCRIPTION

1.1 *2112 RAMs* Each of the eight 2112 RAMs which may be included on the 423 PROM/RAM board contains 1,024 memory cells, organized as 256 four-bit words. To interface with the eight-bit *modular micro* structure, the 2112s are combined into pairs, with one 2112 connected to the four high-order data bus lines on the 423 board (BB7 through BB4), and the other 2112 to the low-order lines (BB3 through BB0). Each pair makes up a *page* in memory--that is, 256 eight-bit words (bytes).

Since each 2112 pair has 256, or 2^8, memory words, eight address lines are required to select which word in a given page is to be written into or read from. Word selection is made by the *modular micro* low-order address bus, DL7 through DL0.

A given *page* in RAM is selected when the chip enable ($\overline{CE}$) terminal on the appropriate pair of 2112s is brought low. On the 423 schematic, the RAM is shown connected as four pages, numbered 010 through 013. The respective chip enable signals are labeled $\overline{CE10}$ through $\overline{CE13}$. (Actually the user may relocate these chips at other page addresses, as described in Sec. 1.3 below.)

The four data input/output terminals of each 2112 RAM use three-state bidirectional bus-driving circuitry. When a 2112 is not chip-enabled, its data terminals are in the high-impedance state, and neither sink nor source any appreciable amount of current. No data transfers can occur between the 2112's memory cells and the bus; some other device on the bus determines bus line voltage levels.

When a 2112 is chip-enabled, its input/output bus drivers are activated. The direction in which they drive is determined by the state of the write enable ($\overline{WE}$) terminal. If this signal is high, the 2112 is in the READ mode; the bus drivers point outwards; and memory data can be read into the CPU. If $\overline{WE}$ is low, the 2112 is in the WRITE mode; the bus drivers point inwards; and the CPU can write data into the selected memory word.

1.2 *1702A PROMs* The 423 also includes room for eight 1702A PROMs. Each device includes 2,048 memory cells organized as 256 eight-bit words. The 1702A, with twice the capacity of the 2112, need not be connected in pairs: each PROM contains a full page of memory. Word selection within each PROM is made by the low-order memory bus, DL7 through DL0. Page selection is accomplished through each PROM's chip enable ($\overline{CE}$) terminals. On the 423 schematic, the PROMs are shown connected as pages 000 through 007, activated by chip-enable signals labeled $\overline{CE0}$ through $\overline{CE7}$. (In practice the user may relocate addresses: Sec. 1.3.)

Like the 2112, the 1702A has three-state bus driver data terminals. However, in the case of the PROM these output terminals are unidirectional only, since the CPU cannot normally write into these devices. The output terminals leave the high-impedance third state when the PROM is chip-enabled, driving the 423's internal data bus, BB7 through BB0.

© 1975 Martin Research. Pp. 1-3, Rev. A; 4, Rev. B; 5-12, Rev. A, Nov. 1975.

SEC. 1 CIRCUIT DESCRIPTION (cont'd)

The 1702A PROM is programmed electrically at the factory with specialized programming equipment. The device retains programmed data indefinitely under normal conditions. Equipped with a quartz window on the top--through which the user can view the silicon chip on which the memory circuitry is etched--the 1702A can be erased by exposure to a strong source of short wave ultraviolet light.

1.3 *Memory Decoding* The 423 PROM/RAM board includes memory decoding circuitry which generates the twelve chip-enable signals for selecting the four pages in RAM and eight pages in PROM. The CPU uses the high-order address bus, DH7 through DH0, to address pages in memory (just as the DL bus selects words within those pages).

Figure 1.3.1 shows how pages are organized into larger blocks of memory. There are four pages in a *K* of memory--that is, 1,024 bytes, or a *kilobyte.* There are 8K in a *bank,* and in a computer using all eight of the DH address lines, a total of eight banks--or 64K words of memory.

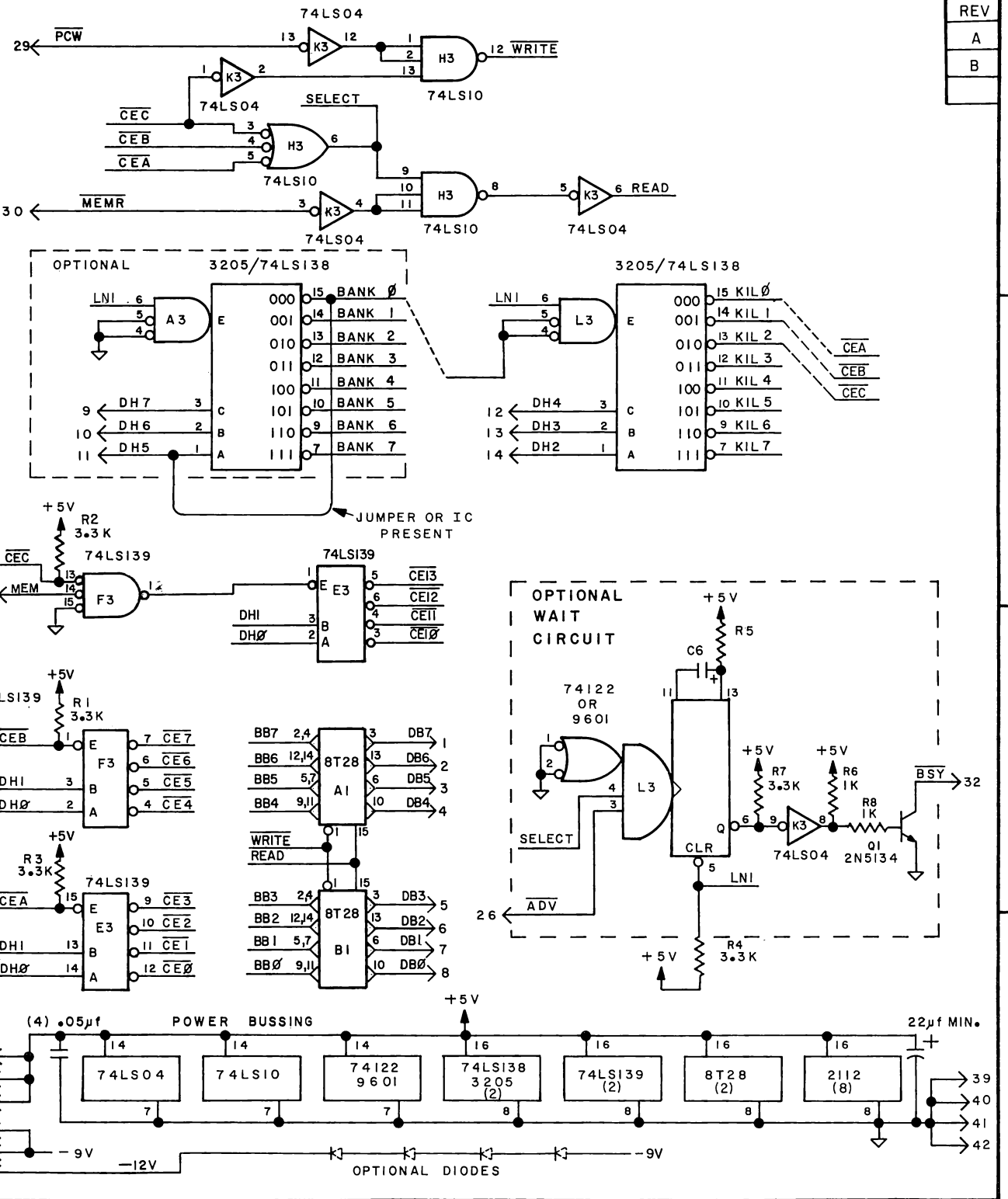
BANK			KIL			PAGE					OCTAL	
DH7	DH6	DH5	DH4	DH3	DH2	DH1	DH0	BANK	K	PAGE	PAGE	COMMENTS
0	0	0	0	0	0	0	0	0	0	0	000	MONITOR location
0	0	0	0	0	1	0	0	0	1	0	004	Fifth PROM begins
0	0	0	0	1	0	0	0	0	2	0	010	Basic RAM
0	0	0	0	1	0	1	1	0	2	3	013	Interrupt RAM
0	0	1	0	0	0	0	0	1	0	0	040	4K RAM, first page
0	0	1	0	1	1	1	1	1	3	3	057	4K RAM, last page
0	1	0	0	0	0	0	0	2	0	0	100	
0	1	1	0	0	0	0	0	3	0	0	140	
1	0	0	0	0	0	0	0	4	0	0	200	
1	0	1	0	0	0	0	0	5	0	0	240	
1	1	0	0	0	0	0	0	6	0	0	300	
1	1	1	0	0	0	0	0	7	0	0	340	
1	1	1	1	1	1	1	1	7	7	3	377	Last page in memory

Figure 1.3.1--Page Addressing

The correspondence between bank/K/page address, and the octal notation commonly used in eight-bit microcomputer software, is also shown in Figure 1.3.1. Octal notation is formed simply by regrouping the address bits--DH7, DH6/ /DH5, DH4, DH3/ /DH2, DH1, DH0--and converting binary bits into octal digits (0 through 7). The DH examples shown include page addresses commonly used in *modular micro* systems.

The 423 decoding circuitry allows the user to address its PROM and RAM in any one of the computer's eight banks. In addition, the positions of PROM and RAM within these banks can be

REV
A
B

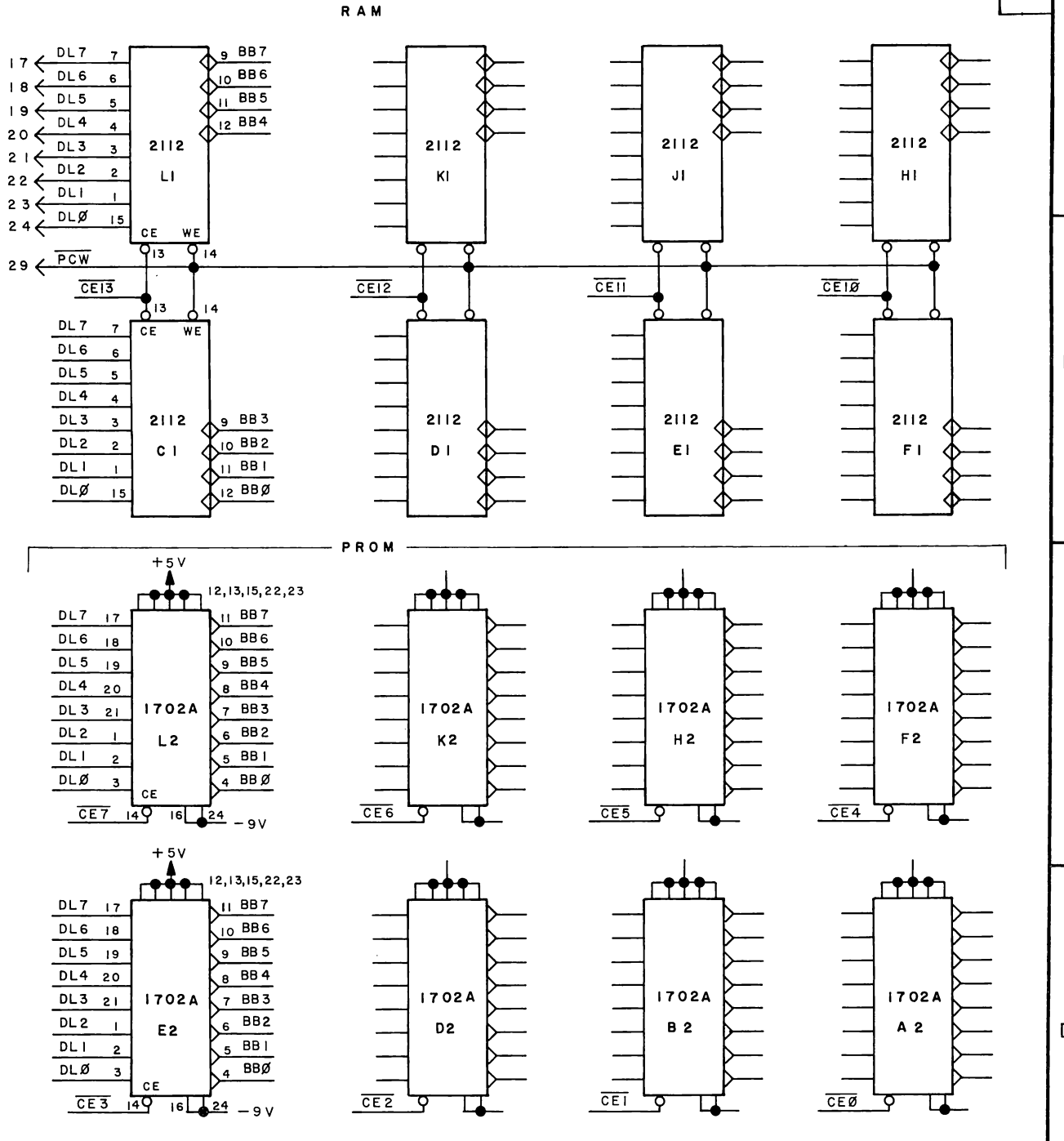


NOTE: PROPRIETARY INFORMATION. NOT TO BE DISCLOSED WITHOUT EXPRESS WRITTEN PERMISSION.	DRAWN BY Roeck	CHECKED BY	DRAWING NO. DS 423 (Page 4)
	DATE 11-20-75	DATE	

EF4A

modular micros
martin research

REV
A



1	2	3
NOTE: PROPRIETARY INFORMATION. NOT TO BE DISCLOSED WITHOUT EXPRESS WRITTEN PERMISSION.	DRAWN BY ROECK DATE 11-20-75	CHECKED BY DATE
		DRAWING NO. DS 423 (Page 5)

EF4A

modular micros martin research

SEC. 1 CIRCUIT DESCRIPTION (cont'd)

selected. The 423's 3K of memory is divisible into three one-kilobyte blocks, one of RAM and two of PROM. Each 1K block can be assigned to any one of the eight possible Ks (or *KILs*) within the selected bank.

In a typical small *modular micro* system, the first four PROMs are assigned to BANK 0, KIL 0; the next four PROMs to BANK 0, KIL 1; and the eight RAMs to BANK 0, KIL 2. This is the way assembled 423 boards are shipped, and these locations appear in the *Comments* column of Fig. 1.3.1.

The 423 board employs several decoders and associated gates to produce the signals required for selecting memory page addresses. The first is the 3205 3-to-8 decoder shown as "optional" (in dotted lines) on the 423 schematic. This chip decodes the DH7, DH6, and DH5 lines to produce eight strobes called $\overline{\text{BANK7}}$ through $\overline{\text{BANKØ}}$. The user must connect a *strap*--that is, a small piece of hookup wire--to select the bank number he desires.

NOTE: The 423-1 PROM/RAM board is sold as part of a *MIKE 2* microcomputer, based on the 8008. This CPU addresses 16K of memory, rather than a full 64K; the DH7 and DH6 lines are grounded on the CPU board. Therefore only two banks in memory are used, banks 0 and 1. The 423-1 is provided with a jumper between DH5 and $\overline{\text{BANKØ}}$ (pins 1 and 15) on the 3205 socket (location A3), effectively selecting BANK0 as the address for the 423 board. To convert the 423-1 for use with other computers, the decoder chip should be added. The extra 3205 is provided with *modular micros* based on the 8080 CPU.

A second 3-to-8 decoder uses the DH4, DH3, and DH2 address lines to produce the eight *KIL* signals, $\overline{KIL7}$ through $\overline{KILØ}$. To assign the 423 board's three one-kilobyte blocks to the desired positions within the selected bank, three straps must be connected between the chosen KIL connections and the points marked $\overline{CEA}$, $\overline{CEB}$, and $\overline{CEC}$. In a typical small *modular micro* system, $\overline{\text{CEA}}$, $\overline{\text{CEB}}$, and $\overline{\text{CEC}}$ are connected to $\overline{\text{KILØ}}$, $\overline{\text{KIL1}}$, and $\overline{\text{KIL2}}$, respectively. $\overline{\text{CEC}}$ is designated for the RAM, and $\overline{\text{CEA}}$ and $\overline{\text{CEB}}$ for PROMs 0-3 and 4-7, respectively.

Two 74LS139 dual two-to-four decoders select pages within each kilobyte. The strobes produced are labeled $\overline{\text{CE0}}$ through $\overline{\text{CE13}}$. Strobes $\overline{\text{CE0}}$ through $\overline{\text{CE7}}$ are used to select the eight PROMs, and strobes $\overline{\text{CE10}}$ through $\overline{\text{CE13}}$, the four pairs of RAM chips. (This page numbering implies that the memory is connected for a small modular *micro system,* in the first 3K of memory; actually, the user may select different page numbers.)

The 423 also generates signals for controlling data transfers on the board. When one of the 423's kilobyte blocks has been selected, $\overline{\text{CEC}}$, $\overline{\text{CEB}}$, or $\overline{\text{CEA}}$ goes low; these are ORed together to produce the signal *SELECT.* $\overline{MEMR}$--a signal produced by the CPU board, indicating that the CPU needs to read memory--is gated with SELECT to produce a *READ* signal. $\overline{\text{PCW}}$, another CPU-produced signal, indicates that the CPU is in a memory write phase. This signal is combined with $\overline{\text{CEC}}$ to produce the signal called $\overline{WRITE}$. READ and $\overline{\text{WRITE}}$ are used to control the bus drivers on the board (Sec. 1.4).

SEC. 1 CIRCUIT DESCRIPTION (cont'd)

A given page in PROM is selected by the appropriate chip enable signal, $\overline{CE7}$ through $\overline{CE0}$. The PROM data appears on the internal data bus, BB7-BB0, and is transmitted to the CPU by the bus drivers when READ goes high.

A given *pair* of 2112s is selected by a chip enable signal, $\overline{CE13}$ through $\overline{CE10}$. Since $\overline{CEC}$ is gated with $\overline{MEM}$ to produce these four chip enable signals, the RAMs are enabled only during CPU cycles referencing memory (read or write). The $\overline{PCW}$ signal is connected to the write enable terminals of all the 2112s, controlling whether the RAM is to be written into or read from.

1.4 *Bus Drivers* The 423 board includes two four-bit bidirectional bus drivers to permit the 2112s and 1702As to drive an expanded *modular micro* data bus. When the signal *READ* is high, the bus-driving outputs shown at the right of the package (in the schematic) are activated, causing the data on the 423's internal data bus, BB7-BB0, to be passed onto the microcomputer data bus, DB7-DB0. At this instant the $\overline{WRITE}$ signal on pin 1 is high, disabling the outputs shown at the left side of the package, and allowing these terminals to be used as inputs.

During memory write cycles, $\overline{WRITE}$ goes low, enabling the inwardly-directed driver section of the bus drivers. The selected 2112 RAM pair can now receive data. Now the READ signal is low, disabling the bus-driving outputs shown at the right of the package (in the schematic), allowing these terminals to be used as inputs. During periods when the 423 board is neither being written into or read from, the bus drivers are entirely inactive, with all terminals in a high-impedance state.

In the 423-1 and 423-2, 423 versions intended for use with 8008 systems, the bus driver used is the 8T28, a non-inverting bus driver. In the 423-2, an 8T26 *inverting* bus driver is substituted. A pin-compatible part, the 8T26 is used to minimize noise problems in large bus structures, and is standard in *modular micro* systems using an inverted data bus--such as those computers based on the 8080 and 6800.

1.5 *Memory Speed* Two factors determine the effective speed at which the 423 PROM/RAM board can run: the speeds of the PROM and RAM chips used; and the mode of memory addressing employed by the CPU in the microcomputer.

The 1702A PROM and 2112 RAM may be characterized either by their *access time*, or by their *select time. Access time* measures the period starting when a valid word address is presented to the memory, and ending when valid data is guaranteed to have been read from (or written into) memory. When using the *select time* spec, it is usually assumed that a valid word address is already available, and the critical period begins when the memory is activated at its chip select terminal, ending when valid data transfer is guaranteed. Normally the select time is appreciably shorter than the access time.

(The time it takes for memory decoding circuitry--like the 3205 or 74LS139--to transfer address signals is also a factor in the system. But this bipolar circuitry is so much faster than MOS memory that these delays are comparatively small.)

SEC. 1 CIRCUIT DESCRIPTION (cont'd)

In a *MIKE 2* microcomputer, the 8008 CPU sends out a DL address first, giving plenty of time for a given word to be selected within a 2112 or 1702A. Then the DH address bits become valid, activating the 423 memory decoder stage, which selects the proper chip through its chip select terminal. Finally, a data transfer occurs between memory and CPU. The period which determines effective memory speed is *select time.*

The 2112 RAM has an access time of 1.0 us and a chip select time of 800ns. The 1702A-6 PROM has an access time of 1.5 us and a chip select time of 900 ns. These chip select times are adequate for full-speed operation in a *MIKE 2* microcomputer operating with an 8008-1 CPU.

In a *MIKE 3* microcomputer, the 8080 CPU provides considerably less gap between the time when an address becomes valid, and the time when a memory read/write may occur. As a result, the effective speed is determined by the memory component's *access time.* In practice, the CPU may have to enter a WAIT state for a slow memory component.

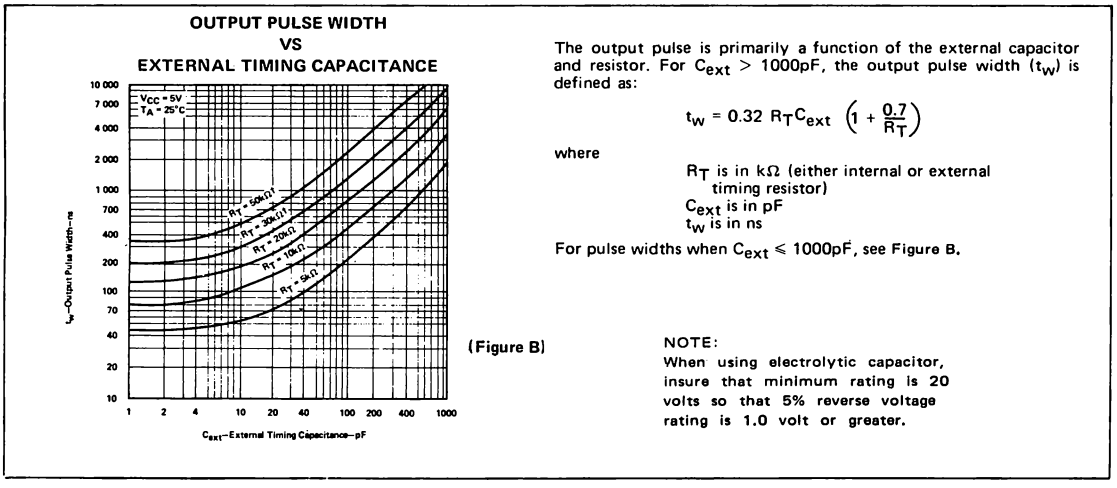
1.6 *Optional WAIT Circuit* When memory components within the microcomputer are insufficiently fast to meet the requirements of the central processing unit, the user must make the CPU *wait* for memory access. In the *modular micro* series, the selected slow memory component must pull down the READY line on the microcomputer bus. Also called $\overline{BSY}$ (indicating that the system is *busy* when the line is low), this line causes the CPU to enter an idle state while the memory component completes its internal word decoding. When the line goes high again, the memory read/write cycle is completed and the CPU continues.

The 423 PROM/RAM board has an optional *WAIT* circuit for accomplishing this function. A one-shot multivibrator, type 74122 or 9601, is triggered only when both (a) a memory location on the board has been selected; *and* (b) the address valid ($\overline{ADV}$) line goes high. This last signal goes low at the time that a memory address is being selected, and when it goes high again, the memory address is stable. It is necessary to gate in $\overline{ADV}$ because the SELECT line may display *glitches* (short pulses) during changes in the DH address register, even though the 423 board is not being referenced. When the one-shot is triggered, its $\overline{Q}$ output goes low for a period of time determined by R5 and C6. Through an inverter, the one-shot turns on a transistor gate, pulling the $\overline{BSY}$ line low.

The 9601 and 74122 one-shots are not identical parts, but either will work in this circuit. Their output pulse widths for a given R and C are virtually identical. Figure 1.6.1 relates R, C, and output timing. The resistor and capacitor chosen should produce a delay appropriate both for the CPU and the memory in use. For example, assume memory access time of 1.0 us and a CPU with a 500 ns cycle time. One CPU memory cycle is not enough to guarantee memory access. The delay is chosen for approximately 700 to 800 ns. This will cause the WAIT delay to end somewhere in the middle of the second CPU memory cycle, allowing it to access memory and continue during its third cycle. This approach avoids the need for close-tolerance parts. The value of R5 should range between 5K and 50K ohms. Suggested values: C6 = 220 pf, 10%. R5 = 8.2K, 10%, when slowest memory is 1.0 us; R5 = 12K, 5% (preferably 5%, but 10% usually is adequate),

SEC. 1 CIRCUIT DESCRIPTION (cont'd)

when slowest memory is 1.5 us. In case of doubt, a larger resistance value should be used. Memory access will be assured, but at a potential loss in system speed (if the CPU loses another memory cycle).



(Figure B)

The output pulse is primarily a function of the external capacitor and resistor. For C_{ext} > 1000pF, the output pulse width (t_W) is defined as:

$$t_W = 0.32\ R_T C_{ext}\left(1 + \frac{0.7}{R_T}\right)$$

where

R_T is in kΩ (either internal or external timing resistor)
C_{ext} is in pF
t_W is in ns

For pulse widths when $C_{ext} \leq$ 1000pF, see Figure B.

NOTE:
When using electrolytic capacitor, insure that minimum rating is 20 volts so that 5% reverse voltage rating is 1.0 volt or greater.

Figure 1.6.1--One-Shot Timing for Selected Values of R and C

Once installed, the WAIT option can be disabled simply by removing the one-shot from its socket. Pull-up resistors associated with the 74LS04 inverter stage will prevent the $\overline{BSY}$ transistor from being turned on.

1.7 *-12 V Option* The user can elect to supply the 423 PROM/RAM board's -9 V requirements from a regulated power supply providing this voltage. Typically the 423 requires 35 MA for each 1702A PROM installed; for conservative power supply rating, allow 40 MA per device. If the user wishes to use a -12 V supply to provide this -9 V requirement, he can add four ordinary silicon power diodes to the 423 board. Connected between the -12 and -9 V lines on the *modular micro* bus, these diodes have forward voltage-drop characteristics of approximately 0.7 volts each, dropping the -12 V to a satisfactory level. (Sec. 3 below illustrates parts location.)

Warning: if the 423, equipped with diodes for -12 V operation, could ever be used in a system where the -9 V is supplied by a -9 V supply, the connection to the -9 V line on the *modular micro* bus should be cut. The place for this cut to be made is indicated in the pictorial in Sec. 3, below.

SEC. 2 PARTS LIST

2.1 *Standard Parts* The following are supplied with the 423-2, a PROM/RAM board with 256 bytes of RAM and 256 bytes of PROM--a single 1702A-6 programmed with the *MIKE 2* Monitor. This is the version of the 423 provided with the *MIKE 2* microcomputer.

Part No.	Quantity	Description
BD423	1	PC BOARD, PROM/RAM
1000	1	CONNECTOR, PC board header, 50-pin, rt. angle solder leads
1012	2	SOCKET, 14-pin DIP, solder-tail
1013	10	SOCKET, 16-pin DIP, solder-tail
1037	1	SOCKET, 24-pin DIP, solder-tail, gold
2005	1	IC, 74LS04 hex inverter
2009	1	IC, 74LS10 triple 3-input NAND
2042	1	IC, 3205 3-to-8 decoder
2043	2	IC, 74LS139 dual 2-to-4 decoder
2130	2	IC, 8T28B quad bi-bus driver
2200-13	1	IC, 1702A-6 256 x 8 PROM, with *MIKE 2* Monitor
2205	2	IC, 2112 256 x 4 RAM, 1.0 us access time
4000-55	4	RESISTOR, 3.3K, 10%, ¼W (color code orange, orange, red)
5501	4	CAPACITOR, .05 to 0.1 uf, 10 V min., ceramic
5502	1	CAPACITOR, 22 to 50 uf, 15 V min., tantalum

2.2 *8080 Version* The following parts are added to (or substituted for) the parts in Sec. 2.1 to produce a 423-4, the PROM/RAM board used in the *MIKE 3* microcomputer, based on the 8080. The differences include an 8080 Monitor PROM; extra memory decoders (Sec. 1.3 above); and an extra page of RAM, page 013. Added parts marked with an asterisk are associated with the optional WAIT circuit (Sec. 1.6).

Part No.	Quantity	Description
1012*	1	SOCKET, 14-pin DIP, solder-tail, tin (for 74122)
2042	1	IC, 3205 3-to-8 decoder (added)
2131	2	IC, 8T26B quad inverting bus driver (replaces P/N 2130)
2205	2	IC, 2112 256 x 4 RAM, 1.0 us (added)
2200-17	1	IC, 1702A 256 x 8 PROM, 1.0 us, with *MIKE 3* Monitor (replaces P/N 2200-13, 1702A-6 with *MIKE 2* Monitor)
3001*	1	TRANSISTOR, 2N5134 NPN (Q1)
4000-49*	2	RESISTOR, 1K, 10%, ¼W (color code brown, black, red) (R6, R8)
4000-55*	1	RESISTOR, 3.3K, 10%, ¼W (color code orange, orange, red) (R7)
4000-59*	1	RESISTOR, 6.8K, 10%, ¼W (color code gray, red, red) (R5; for 1 us memory)
5000-21*	1	CAPACITOR, 500pf, 10%, 1 KV, ceramic (C6)

SEC. 2 PARTS LIST (cont'd)

2.3 *Memory Expansion* The 423 can be expanded to its full capacity by adding suitable RAMs and PROMs. In addition to the unprogrammed PROMs listed below, programmed devices including useful software are available from Martin Research. See the current catalog and Software Package data sheets. Prices for parts: see the Replacement Parts List.

Part No.	Quantity	Description
1013	- - - - -	SOCKET, 16-pin DIP, solder-tail, tin (for RAMs)
1037	- - - - -	SOCKET, 24-pin DIP, solder-tail, gold (for PROMs)
2200	- - - - -	IC, 1702A-6 256 x 8 PROM, 1.5 us access time (unprog.)
2205	- - - - -	IC, 2112 256 x 4 RAM, 1.0 us access time
2209	- - - - -	IC, 1702A 256 x 8 PROM, 1.0 us access time (unprog.)

2.4 *Optional WAIT Circuit* The optional WAIT circuit (described in Sec. 1.6 above) requires the parts marked with asterisks in Section 2.2 above plus the following:

Part No.	Quantity	Description
2037	1	IC, 74122 one-shot (a 9601 one-shot may be substituted)

2.5 *-12 V Option* To provide the -9 V requirement from a -12 V supply:

Part No.	Quantity	Description
3702	4	DIODE, SD1, 1A, 50V (or equivalent)

SEC. 3 ASSEMBLY NOTES

3.1 *Parts Locations* Figure 3.1.1 illustrates the locations of the RAM, PROM, optional WAIT, and -12 V circuitry on the 423 board. The component side of the board is shown.

REV. A

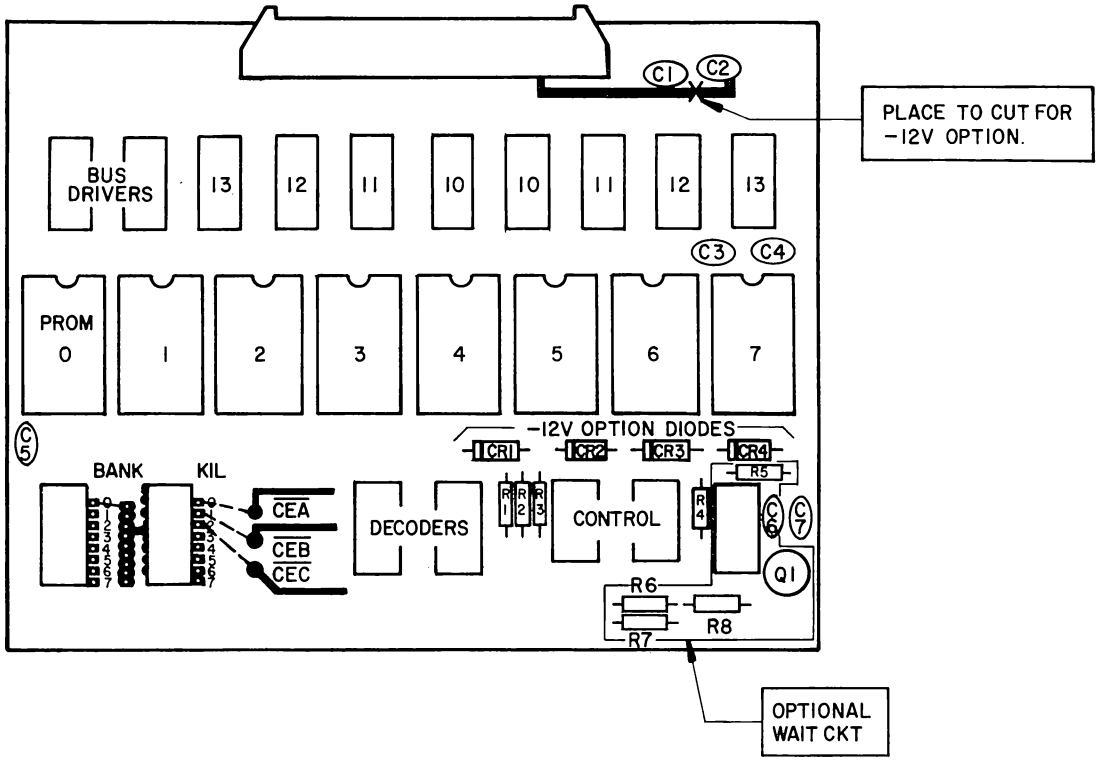


Figure 3.1.1--Parts Locations on the 423 Board

The *415 Breadboard* is a printed circuit board for use in interfacing custom electronic circuitry to a modular microcomputer system. Useful in prototyping and for the experimenter, the 415 is a standard-size modular micro board with over seven hundred pre-tinned holes, arranged to accomodate standard integrated circuits. ICs may be inserted directly into the board, or sockets may be plugged into the board, interconnections being made point-to-point with hookup wire and solder. For greatest flexibility, accessory kits of sockets with long square leads are available, for use with wirewrap tools.

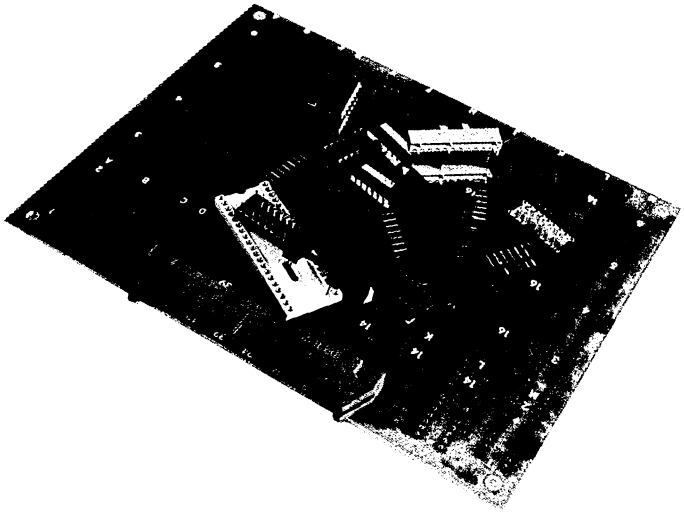
The 415 connects to the modular micro bus via a standard fifty-pin connector. The 415 provides a convenient place for signals and power supply connections external to the modular micro system to be brought onto the computer bus.

A number of special circuit kits are available which allow the user to add special functions to the modular microcomputer system. Designed for insertion into a 415 breadboard, in general these include full instructions and a collection of parts and sockets. Numbered in the 9000 series, these kits are described in other product literature from Martin Research.

The 425 breadboard is an updated version of the Model 415 and will be supplied when new orders are received. The 425 differs only in having extra holes on either side for installing 20- to 50-pin right angle headers, allowing cable connections to accessories and peripheral devices.

The *425 Breadboard* is a printed circuit board for use in interfacing custom electronic designs to a modular microcomputer system. Useful in prototyping and for the experimenter, the 425 is a standard-size modular micro board, with over nine hundred pre-tinned holes, arranged to accommodate standard integrated circuits. ICs may be inserted directly into the board, or sockets may be used, interconnections being made point-to-point with hook-up wire and solder. Or, for greater flexibility, accessory kits of sockets for use with wirewrap tools are available.

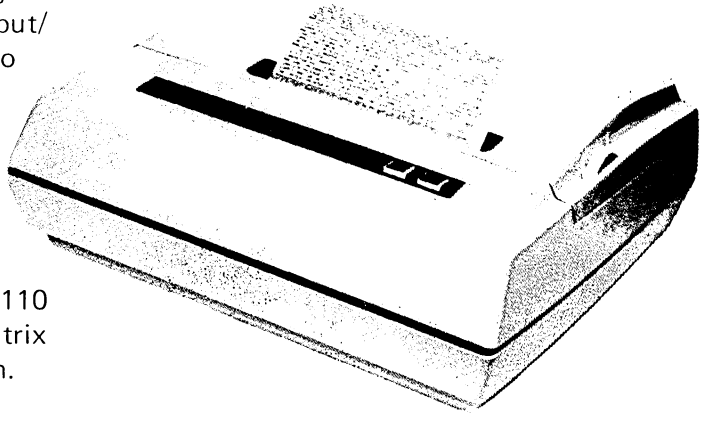
The 425 connects to the modular micro bus with a standard fifty-pin connector. The 425 provides a convenient place for signals and power supply connections external to the modular micro system to be brought onto the computer bus. The 425 also includes space on either side for additional flat cable headers (20/40/50-pin), for interfacing via cable to accessories and peripheral devices.



BREADBOARD
with assortment of
sockets for custom circuitry

A number of circuit kits are available which allow the user to add special functions to the modular microcomputer system. Designed for insertion into a 425 breadboard, in general each includes a collection of parts and sockets, and full instructions.

Available now: kits for interfacing to ASCII keyboards; to 8-bit parallel TTL input/output sources; to the ASR33; and to other makes of printers.



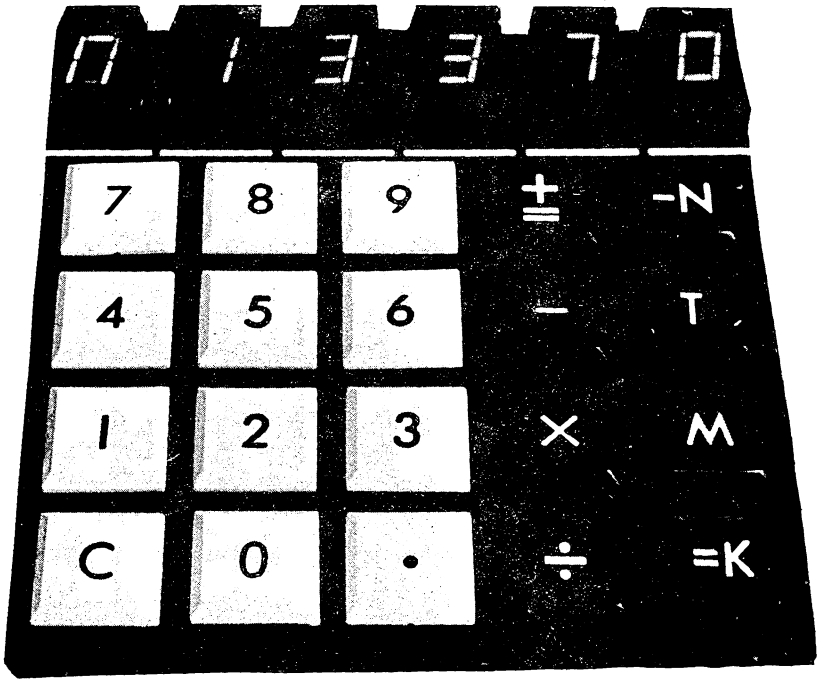
Shown at right: The PR11000, a 110 character per second, 80-column dot matrix printer available from Martin Research. Write for details, pricing.

The *420 Console Board* consists of a calculator-type keyboard and a six-digit display. In a typical modular micro system, the keyboard is used to program the microcomputer, and the bright, large (0.3-inch) digits are used to display data and memory addresses.

The keyboard may also be used for any other data input function, and the digits for any other display purpose--under control of the microcomputer. The keyboard is simply an input port, and the display, a set of output ports. The digit drivers will display data in octal, decimal, or hexadecimal (0-9 and A-F) formats.

A significant advance over older console designs--where banks of switches and lights are hardwired into the system control circuitry--the 420 keyboard/display reflects state-of-the-art systems design, as used in the latest generation of small computers.

PHOTO: Closeup shows bright digits, 20 keys



SPECIFICATIONS

Keyboard interface:

The numerals 0-9, and six special-function keys, are electronically interpreted in hexadecimal and make up microcomputer input port 006. Two more control keys interface as additional bits in input 006. The remaining two keys are related to the system interrupt function.

Display interface: The six LED digits are driven by microcomputer output ports 015, 016, and 017. The 423's decoders display the numerals 0 through 9, and the hexadecimal numbers from A through F, thus: A, b, C, d, E, F.

System interface: Board connects directly to the *modular micro* bus with standard fifty-pin connectors. Includes bus drivers for adequate drive power in expanded systems.

Power: Requires a single +5 volt supply, ±5%, 950 MA.

REV. A

TECHNICAL DATA

SEC. 1 CIRCUIT DESCRIPTION

1.1 *Console Board Structure* The 420 consists of three microcomputer output ports, driving six seven-segment digits, and a single microcomputer input port, used to scan a 20-pad keyboard. The 420 also includes an interrupt latch for interrupting the microcomputer; a simple digital probe (in the *B* revision); and additional circuitry necessary for interfacing with the microcomputer.

The keyboard is a calculator-style device which includes the ten decimal digits 0-9, plus ten function keys, as shown on the first page of this data sheet. Two keys (*-N* and *T*) are dedicated to the interrupt function. The *Monitor* program provided with a typical *modular micro* system allows the user to program the microcomputer with the 420 keyboard, and to inspect data on the 420's six digits. The *Operating Instructions* that come with the system explain the significance of each key in programming the computer.

However, each key may take on whatever significance the user desires, depending on how the microcomputer has been programmed. And, the six digits may also be used for any desired display function. In short, neither the keyboard nor the display is permanently wired into the CPU--as would be typical in computers featuring a front panel with switches and lights which are used only for programming and for displaying CPU status. The 420 Console Board is a data entry and display device whose complete flexibility is in keeping with the *modular micro* concept.

1.2 *Digital Outputs* The 420's digits make use of light-emitting diode (LED) technology. Each digit is made up of seven segments, which are turned on when supplied with +2 V. The common terminal of each digit is returned to ground (e.g., the MAN-74 is a common-cathode device). The digit's seven terminals are controlled by a type 9368 hex-to-7-segment decoder, which receives a four-bit (hex) code and causes the appropriate segments of its respective digit to light up. The 9368 includes internal current regulation, supplying each activated segment with approximately 20 milliamps of current.

The 9368 converts four-bit hexadecimal code into a visual display as follows: 0000 = 0; 0001 = 1; 0010 = 2; 0011 = 3; 0100 = 4; 0101 = 5; 0110 = 6; 0111 = 7; 1000 = 8; 1001 = 9; 1010 = A; 1011 = B; 1100 = C; 1101 = D; 1110 = E; 1111 = F. The letters *A* through *F* are hex for the decimal values 10 through 15.

The six digits are arranged in three pairs. Digits 5 and 4 are addressed by the microcomputer as Output 017; digits 3 and 2 as Output 016; and digits 1 and 0 as Output 015. Within each output port, the four high-order bits of the output data word are assigned to the higher-numbered digit. The four low-order bits determine the lower-numbered digit.

8	7	6	5	4	3	2	Rev	ECO	By	Ck	Date
A	A	A	A	A	A	A	A	016	KB	BR	12·4·75

©1975, Martin Research, Northbrook, Illinois 60062 USA. All rights reserved.

SEC. 1 CIRCUIT DESCRIPTION (cont'd)

The output ports are connected to the microcomputer's low-order address bus, DL7-DL0. When for example an OUTPUT 017 instruction is executed, the output data for digit 5 appears at DL7-DL4, and the output data for digit 4 appears at DL3-DL0. When the Output 017 strobe --DT17--goes low, these values are entered into the 9368's internal latches. The 9368 retains this information after the strobe pulse goes away, changing the values displayed by digits 5 and 4 only when a new Output 017 instruction is executed.

1.3 *Sixteen Scanned Keys* The keyboard consists of twenty keyswitches. The microprocessor scans sixteen of these keys using a combination of hardware and software.

The four low-order bits of the microcomputer address bus, DL3-DL0, are connected to the inputs of a 74154/9311 four-to-sixteen decoder. There are sixteen possible combinations of these four bits--0000 (*0* in octal), 0001 (*1* in octal), etc., through 1111 (*17* in octal). One of the 74154's sixteen output terminals is always low, depending on the states of DL3-DL0. The output terminals within the 74154 outline (on the schematic) show the appropriate octal code.

When a key is depressed, a connection is made between a signal line and a common terminal. Sixteen of these signal lines (shown at the left of the keyboard in the 420 schematic) are connected to the 74154 decoder outputs. If a given key has been depressed-- *and* if that key has been selected by the decoder--a connection is made between the 74154 output and the keyboard common terminal. Normally held to +5 volts by a 3.3K pullup resistor, the common terminal is now pulled to a logic low level, near 0 volts.

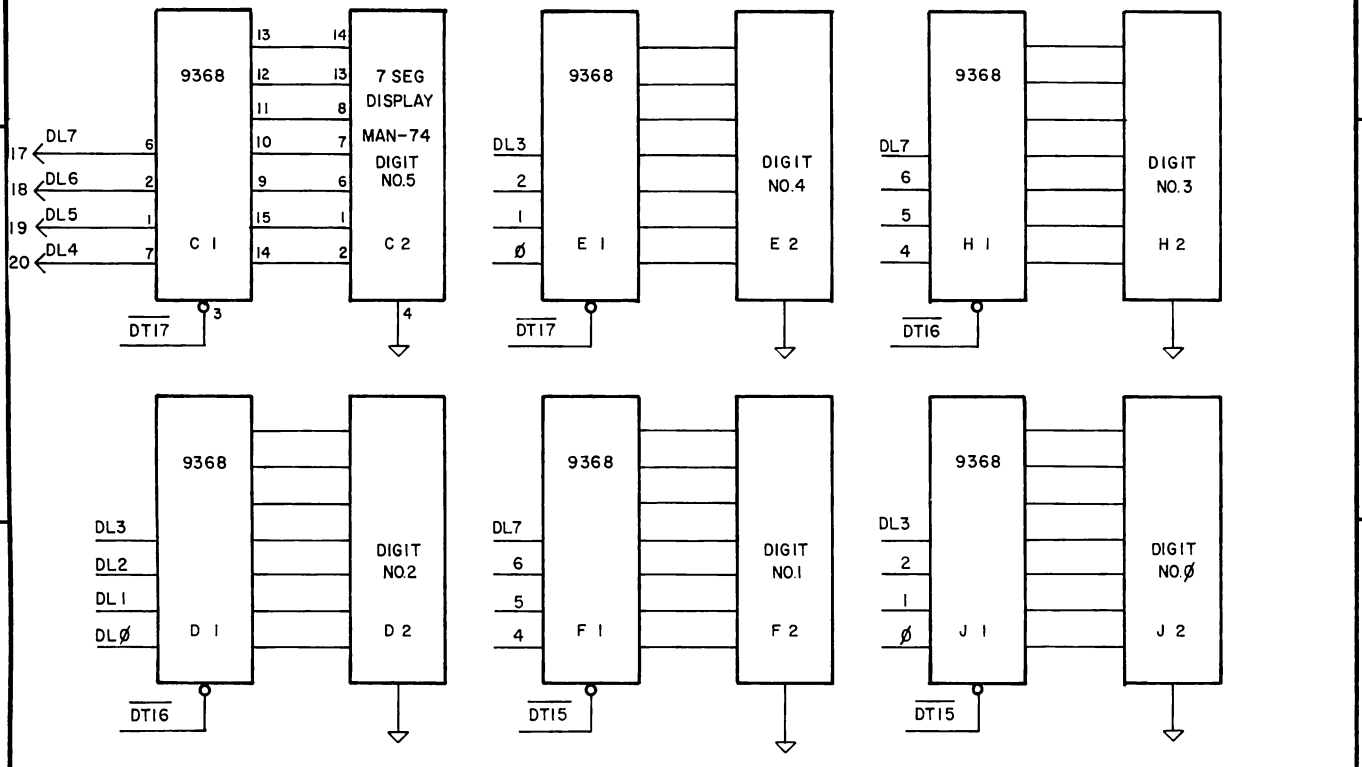
To read the keyboard, the microcomputer executes an INPUT 006 instruction. This creates a DN6 enable signal, which turns on a single three-state bus driver--part of an 8T97B. The keyboard data (*KBD* line) is now impressed onto the low-order bit of the microcomputer data bus, DB0, and is read into the CPU. If this bit is a logic one, the key selected by DL3-DL0 was not depressed; if it is a zero, the key was turned on.

The microcomputer is programmed to scan the sixteen keys repetitively. First the low-order memory address is set up, to address the desired key. Then an INPUT 006 instruction is executed, to read whether that key has been depressed. If not, the low-order address is incremented and the procedure is repeated. If the key was depressed, then suitable action can be initiated under program control. In practice, a delay must be included in order to allow for keyswitch contact bounce. For an example of suitable programming, see description of the *Monitor* program provided with the *modular micro* system.

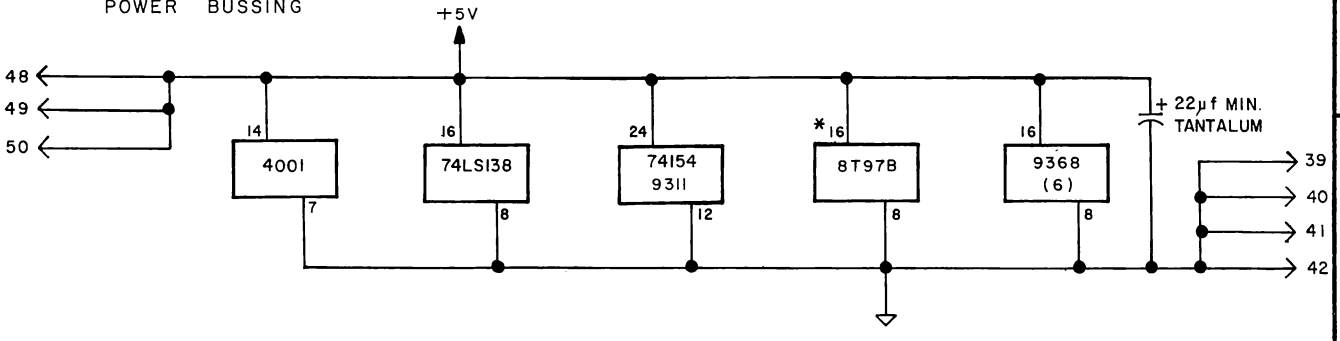
The sixteen keys scanned in this manner are shown at the lefthand side of the keyboard, on the 420 schematic. They include the labels C, ·, ÷, −, and ±, and the digits 9 through 0. The four-bit code on the address register for any one of the number keys is simply that number in binary; this simplifies the programming required to interpret key closures.

REV
A

DISPLAY



POWER BUSSING



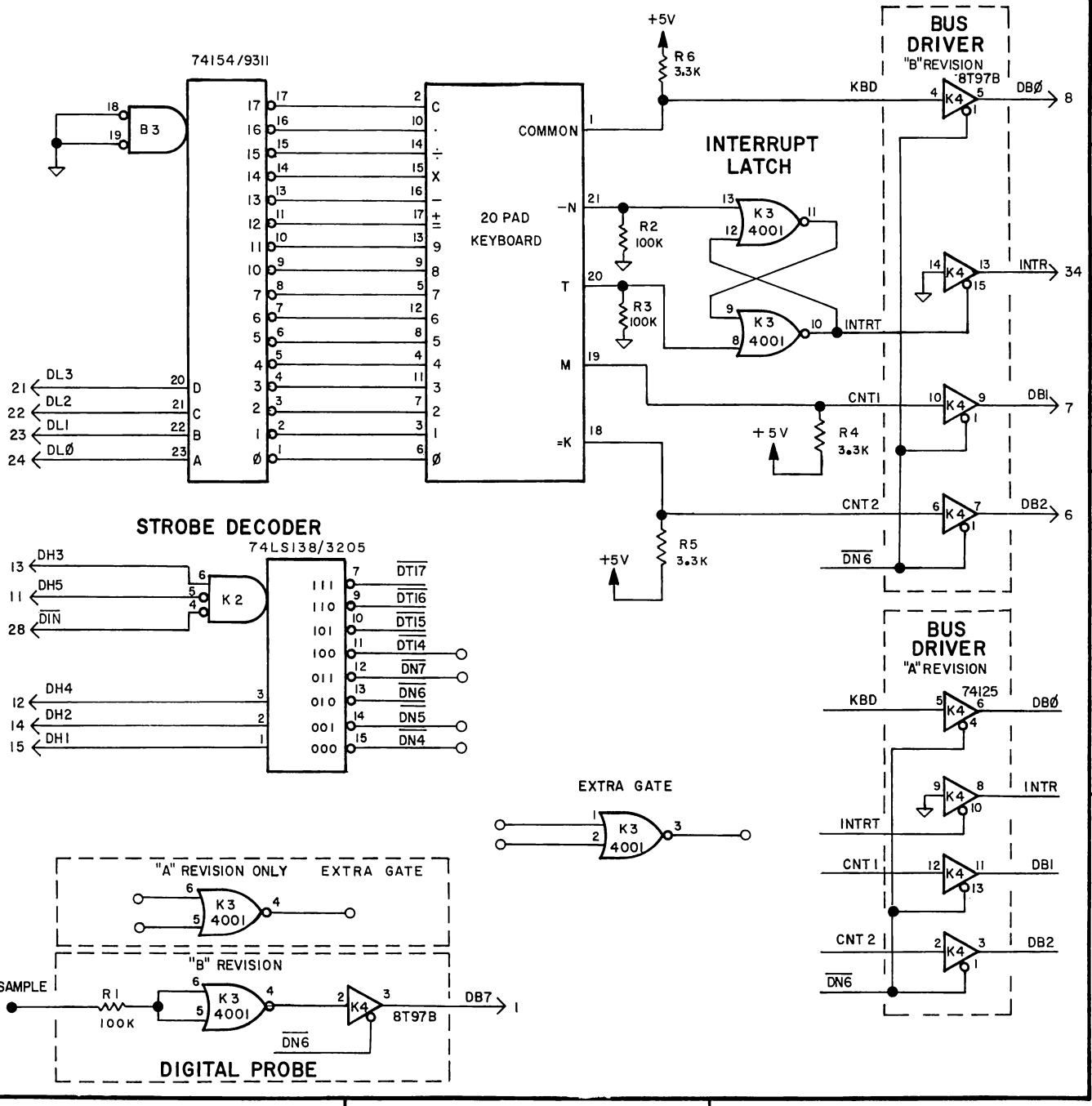
* "A" REVISION 74125

NOTE: PROPRIETARY INFORMATION. NOT TO BE DISCLOSED WITHOUT EXPRESS WRITTEN PERMISSION.	DRAWN BY Schneider	CHECKED BY D.K	DRAWING NO. DS 420 (Page 4)
	DATE 11-19-75	DATE 11-17-75	

EF4A

modular micros martin research

REV
A



1	2	3
NOTE: PROPRIETARY INFORMATION. NOT TO BE DISCLOSED WITHOUT EXPRESS WRITTEN PERMISSION.	DRAWN BY Schneider DATE 11-19-75	CHECKED BY DTL DATE 11-19-75	DRAWING NO. DS 420 (PAGE 5)

EF4A

modular micros martin research

SEC. 1 CIRCUIT DESCRIPTION (cont'd)

1.4 *Shift Keys, M, =K* The two keys *M* and *=K* are shift (or control) keys. They may be used together with the sixteen scanned keys to create new input codes, in much the same way as the shift keys on a typewriter. For the microcomputer to read one of the shift keys--say *M*--any one of the sixteen scanned keys must also be depressed. This pulls the keyboard common terminal to logic zero. Since the *M* key is also depressed, connecting the M output to common, the keyboard output labeled *M* is also low. This signal, shown as *CNT1* (control one), is connected to an extra bus driver. The same INPUT 006 instruction which is used to scan the sixteen keys also inputs this control key, with the input bit appearing on DB1 data bus line. In the same way, the *=K* key creates the signal *CNT2,* appearing on the DB2 data bus line.

These shift keys effectively multiply the number of available key functions. Each of the sixteen keys may be read by the CPU in four different ways: with neither control key pressed (the normal mode); with only the *M* key pressed; with only the *=K* key pressed; and with both control keys pressed. The proximity of the two shift keys makes it easy to depress both at the same time.

1.5 *Interrupt Latch* The *-N* and *T* keys are not scanned by the CPU, but instead are used for interrupting the microcomputer. The T key is a PRESET INTERRUPT key, and the -N key, an INTERRUPT key. When T is pressed, the interrupt latch pulls the *INTR* line on the *modular micro* bus to a low level; when -N is pressed, the latch is reset, allowing INTR to go high, interrupting the CPU on the rising edge.

Both the -N and T lines are normally held to logic low levels by 100K ohm resistors to ground, R2 and R3. When a key is depressed, a connection is made to the common line, held to +5 volts by R6. A logic high level at -N triggers the upper 4001 NOR gate. Two gates are cross-coupled as a set-reset flip-flop, with -N setting the latch, and T resetting the latch. When the flip-flop's *Q* output, *INTRT,* goes low, it activates an 8T97 bus driver stage. This driver leaves its high-impedance state, and since its input is grounded, pulls the *INTR* line low. This open-collector line, with a pullup resistor on the CPU board, causes an interrupt when the flip-flop is reset and INTR goes high. (This circuit provides an interesting example of how to use a tri-state driver in an open-collector circuit.)

1.6 *Input/Output Addressing* The 420 board uses one of the several modes of input/output addressing which are available in the *modular micro* series. Essentially conforming to the I/O addressing method used by the 8008 microprocessor, I/O selection is also compatible with central processor boards offered by Martin Research which use the 8080 and 6800 microprocessors.

The input/output port address appears on five bits of the high-order memory address bus, DH5 through DH1. The I/O port number ranges from 00 to 37, the thirty-two octal numbers which result from the possible combinations of five binary bits. A *strobe decoder* stage on the 420 board combines a universal I/O strobe/enable signal, $\overline{\text{DIN}}$--which is created by the CPU board --with the address signals DH5 through DH1 to create the needed signals for addressing I/O ports

SEC. 1 CIRCUIT DESCRIPTION (cont'd)

on the board. These are $\overline{DT17}$, $\overline{DT16}$, and $\overline{DT15}$--for the digital display output ports-- and $\overline{DN6}$, for the keyboard input port.

When $\overline{DT17}$ goes low (for example), valid information for Output 017 exists on the low-order address bus, DL7-DL0. The 9368s for digits 5 and 4 accept this information, as explained above. When $\overline{DN6}$ goes low, the CPU is ready to accept information from Input 006 on its data bus, DB7-DB0. The bus drivers are enabled and transmit this information to the CPU, as detailed earlier.

1.7 *Bus Drivers* The three-state bus driver IC used in the 420 board is a type 8T97, except in the *A* version, where a 74125 is used. Operation of the two types is similar: both input and output terminals are in a high-impedance state until the enable terminal (drawn with a bubble) is brought to a logic low level. At that point, the output circuitry takes on the voltage level of the inputs. The 74125 package includes four bus drivers with separate enable terminals; the 8T97 includes six drivers with a common enable signal, different pinouts, and increased driving power. The extra driver in the *B* version permits the addition of a simple digital probe.

1.8 *Digital Probe* The 420 *B* version makes use of one of the extra 8T97 bus driver stages by combining it with an extra 4001 NOR gate to produce a simple digital probe. An input signal of interest may be connected to the test point, marked *TP* on the center right side of the 420B console board. This signal is coupled through a 100K ohm resistor and a 4001 inverting buffer stage to an extra 8T97 bus driver. When an INPUT 006 instruction is executed, this value is transmitted as the high-order data bit, DB7. The CPU can test the state of this bit in order to perform the desired test function. The 100K resistor provides a high-impedance probe, unlikely to overload any digital circuitry. Typically, input voltages of 0 to 1.5 volts will result in a *high* DB7 bit, and inputs of 3.5 to 5.0 volts will result in a *low* DB7 bit. Intermediate voltages, falling within the transitional range of the 4001, may produce unpredictable results. Input levels should not exceed +5 volts.

SEC. 2 PARTS LIST

2.1 *B Version* The following parts are supplied with the 420 in the *B* revision.

Part No.	Quantity	Description
BD420	1	PC BOARD, Console
1000	1	CONNECTOR, PC board header, 50-pin, rt. angle solder leads
2042	1	IC, 74LS138/3205 3-to-8 decoder
2050	1	IC, 74154/9311 4-to-16 decoder
2132	1	IC, 8T97B hex bus driver
2150	6	IC, 9368 hex LED digit driver
2401	1	IC, 4001 quad NOR gate, CMOS

SEC. 2 PARTS LIST (cont'd)

Part No.	Quantity	Description
3502	6	LED, MAN-74 7-segment digit
4000-55	3	RESISTOR, 3.3K, 10%, ¼w
4000-73	3	RESISTOR, 100K, 10%, ¼w
5502	1	CAPACITOR, 22 to 50 uf, 15 V min., tantalum
6500	1	KEYBOARD, 20 keys, calculator style, on PC board

2.2 *A Version* The *A* version of the 420 board is identical except as noted:

Part No.	Quantity	Description
2041	1	IC, 74125 hex tri-state driver (instead of P/N 2132, 8T97B)
4000-55	2	RESISTOR, 100K, 10%, ¼w (instead of 3 of same; R1 deleted)

The *Model 9001 Universal I/O Kit* includes the components and IC sockets necessary for constructing an eight-bit TTL input port and an eight-bit TTL output port. For construction on a *modular micro* breadboard, this kit is a general-purpose I/O device which serves as a useful introduction to microcomputer interfacing.

SPECIFICATIONS

I/O Addressing	Input and output port address may be selected by the user. Details are provided in this data sheet.
Bus Driver Power	Three-state bus drivers draw less than 40 uA from each data bus (DB) line when inactivated, and are capable of driving up to 48 MA (30 TTL loads) when selected as an input port.
Output Port Loads	The output port imposes one standard TTL load on the low-order address bus, DL7-DL0. Output port can drive up to 10 TTL loads.

TECHNICAL DATA

SEC. 1 CIRCUIT DESCRIPTION

1.1 *Inputs and Outputs* Communications between a microprocessor and data-handling circuitry external to the central processing unit are commonly implemented by executing *input* and *output* (I/O) instructions. The CPU board generates timing signals necessary for synchronizing the I/O device with the CPU--a *strobe* signal which causes an *outport port* to latch up information on a bus at the appropriate time, or an *enable* signal which activates the bus-driving circuitry associated with an *input port,* causing the input data to be transmitted to the CPU.

Just as when the CPU is reading from or writing into memory, a means must be provided for *addressing* the desired I/O port. This I/O port address must be available outside the CPU board at the same time that the I/O data transfer takes place--in practice, for proper synchronization, a little earlier.

Examples of I/O ports are printers; keyboards; television typewriters and graphic displays; analog to digital converters and digital to analog converters; digital displays; cassette recorders; and so on.

1.2 *I/O Addressing Methods* There are at least three common modes for addressing I/O ports, each being used by a different standard eight-bit microprocessor. In the *modular micro* series, where the bus structure is compatible with a number of central processing units, one or several of these I/O address modes may be used--depending on the CPU that has been installed. Although the fine details are provided in data sheets concerning the respective central processor boards--based on the 8008, 8080, and 6800--the following serves as a brief introduction.

SEC. 1 CIRCUIT DESCRIPTION (cont'd)

8008 Mode--In the 8008, the simplest of these processors, the I/O port address appears on five bits of the high-order address bus, namely DH5 through DH1. *Output* data appears on the low-order address register, DL7-DL0. *Input* data is transmitted to the CPU via the CPU data bus, DB7-DB0. The CPU board develops a universal I/O strobe/enable signal called $\overline{\text{DIN}}$, which is used to activate data transfers between the CPU and I/O ports.

8080 Mode--In the 8080, the I/O port address appears in duplicate on both address busses, DH7-DH0 and DL7-DL0. Both *input* and *output* data transfers take place via the CPU data bus, DB7-DB0. The 8080 microprocessor provides signals which are used to create I/O strobes for activating these data transfers at the appropriate time. Data transfer is unidirectional--i.e., *input* only or *output* only.

6800 Mode--In the 6800, there are no I/O instructions as such; I/O ports are simply addressed as memory. Typically one section of memory is set aside for I/O use. Just as with memory, I/O data transfers take place via the data bus, DB7-DB0.

Central processor boards based on the 6800 and 8080 can be made compatible with 8008-mode I/O addressing by designating certain memory addresses for I/O ports, and by developing the $\overline{\text{DIN}}$ strobe/enable signal on the 6800 or 8080 CPU board. This is the approach used in the *modular micro* series. However, the fact that 8008-compatible I/O instructions are possible in a *modular micro* system based on the 6800 or 8080 does not mean that the system *must* use this I/O addressing mode. The CPU's basic addressing mode can still be used. The only restriction is that a certain memory address not be used for conflicting purposes. As with *any* microcomputer, the user must maintain a *memory map,* to ensure that a given address is assigned once and only once.

1.3 *The 32 Compatible I/O Instructions* Since five address lines are used in the 8008 mode of I/O addressing, there are 32 possible I/O instructions. These are shown in Figure 1.3.1.

At the left of Fig. 1.3.1 appear the five I/O address bits, DH5 through DH1. When the CPU board in use is based on the 8008 microprocessor, these 32 I/O ports are addressed by *input* or *output* instructions, as indicated, ranging from *INP 00* to *OUT 37.* The octal codes for these 8008 instructions are shown in an adjoining column. These codes result from the five I/O address bits, DH5-DH1, *and* from the following conditions: *CC2* (DH7) = 0, CC1 (DH6) = 1, and DH0 = 1. These codes happen to be the values which are microprogrammed into the 8008 microprocessor for I/O instructions.

© 1975 Martin Research, Northbrook, IL 60062 USA. All rights reserved.

8	7	6	5	4	3	2	Rev	ECO	By	Ck	Date
A	A	A	A	A	A	A	A	016	KB	BL	12-5-75

REV. A

SEC. 1 CIRCUIT DESCRIPTION (cont'd)

I/O ADDRESS BITS					8008 INSTRUCTION		EQUIVALENT PAGE 8080		6800	
DH5	DH4	DH3	DH2	DH1	MNEMONIC	OCTAL CODE	OCTAL	HEX	OCTAL	HEX
0	0	0	0	0	INP 00	101	301	C1	201	81
0	0	0	0	1	INP 01	103	303	C3	203	83
0	0	0	1	0	INP 02	105	305	C5	205	85
0	0	0	1	1	INP 03	107	307	C7	207	87
0	0	1	0	0	INP 04	111	311	C9	211	89
0	0	1	0	1	INP 05	113	313	CB	213	8B
0	0	1	1	0	INP 06	115	315	CD	215	8D
0	0	1	1	1	INP 07	117	317	CF	217	8F
0	1	0	0	0	OUT 10	121	321	D1	221	91
0	1	0	0	1	OUT 11	123	323	D3	223	93
0	1	0	1	0	OUT 12	125	325	D5	225	95
0	1	0	1	1	OUT 13	127	327	D7	227	97
0	1	1	0	0	OUT 14	131	331	D9	231	99
0	1	1	0	1	OUT 15	133	333	DB	233	9B
0	1	1	1	0	OUT 16	135	335	DD	235	9D
0	1	1	1	1	OUT 17	137	337	DF	237	9F
1	0	0	0	0	OUT 20	141	341	E1	241	A1
1	0	0	0	1	OUT 21	143	343	E3	243	A3
1	0	0	1	0	OUT 22	145	345	E5	245	A5
1	0	0	1	1	OUT 23	147	347	E7	247	A7
1	0	1	0	0	OUT 24	151	351	E9	251	A9
1	0	1	0	1	OUT 25	153	353	EB	253	AB
1	0	1	1	0	OUT 26	155	355	ED	255	AD
1	0	1	1	1	OUT 27	157	357	EF	257	AF
1	1	0	0	0	OUT 30	161	361	F1	261	B1
1	1	0	0	1	OUT 31	163	363	F3	263	B3
1	1	0	1	0	OUT 32	165	365	F5	265	B5
1	1	0	1	1	OUT 33	167	367	F7	267	B7
1	1	1	0	0	OUT 34	171	371	F9	271	B9
1	1	1	0	1	OUT 35	173	373	FB	273	BB
1	1	1	1	0	OUT 36	175	375	FD	275	BD
1	1	1	1	1	OUT 37	177	377	FF	277	BF

Figure 1.3.1--The 32 Compatible I/O Instructions

SEC. 1 CIRCUIT DESCRIPTION (cont'd)

For a distinction between 8008 *input* and *output* instructions, see Sec. 1.7 below.

When a *modular micro* CPU board based on the 8080 or 6800 is in use, these 8008-compatible I/O ports are addressed as pages in memory. In an *8080 computer,* the following conditions apply to the memory addresses which are assigned as I/O ports: DH7 = DH6 = DH0 = 1. In a *6800 computer,* the conditions are: DH7 = 1; DH6 = 0; DHO = 1. These address conditions are chosen to fit in with the memory organization typically used with each CPU. Each CPU board uses the appropriate equation in creating the $\overline{\text{DIN}}$ signal. The memory page numbers which result are shown in Fig. 1.3.1, in both octal and hexadecimal notations.

1.4 *I/O Strobe Decoding* In order to synchronize an I/O port to the CPU, it is necessary to develop strobe/enable signals. This is done by combining the universal I/O signal created by the CPU board--$\overline{\text{DIN}}$--with the five I/O address lines, DH5 through DH1.

Rather than combining the DH register bits with $\overline{\text{DIN}}$ on the CPU board itself, and piping a number of I/O signals around the system, the *modular micro* system uses a flexible *peripheral strobe decoding technique.* A single 74LS138 or 3205 3-to-8 decoder, installed on any peripheral circuit board, creates eight strobe/enable signals by combining $\overline{\text{DIN}}$ and the five I/O address bits. By connecting these signals to the 'LS138 in different manners, any one of twenty different combinations of I/O signals can be created. See Figure 1.4.1.

The 'LS 138 has eight active-low output terminals. Only one can be low at any given time. The chip's three input terminals--*C, B,* and *A*--are used to select outputs. For example, when C, B, and A are high (input code 111), the output terminal marked *7* (within the package outline in Fig. 1.4.1) is selected. (*7* is octal for *111.*) An additional condition to produce an active-low output signal is that the chip enable terminals--E3, E2, and E1--all be in their active states. E3 must be high, and E2 and E1 (the bubbled enable inputs) must be low.

In practice, one of the active-low enable terminals is assigned to $\overline{\text{DIN}}$. Only when this universal I/O signal goes low, can any I/O strobe/enable signal be produced. This leaves five 'LS138 inputs for connection to the I/O port address lines, DH5 through DH1. A comparison of Figures 1.3.1 and 1.4.1 should demonstrate that the 'LS138 produces the signals shown.

Note that with this scheme, INP 00 is never decoded. This port would be addressed when all five DH bits were logic zero; but the 'LS138 requires that there be at least one active-high input bit (at pin 6). For a similar reason, the $\overline{\text{DT37}}$ strobe cannot be generated. Therefore I/O ports 00 and 37 cannot be selected with just the 'LS 138. Adding one or two inverters would remove this limitation.

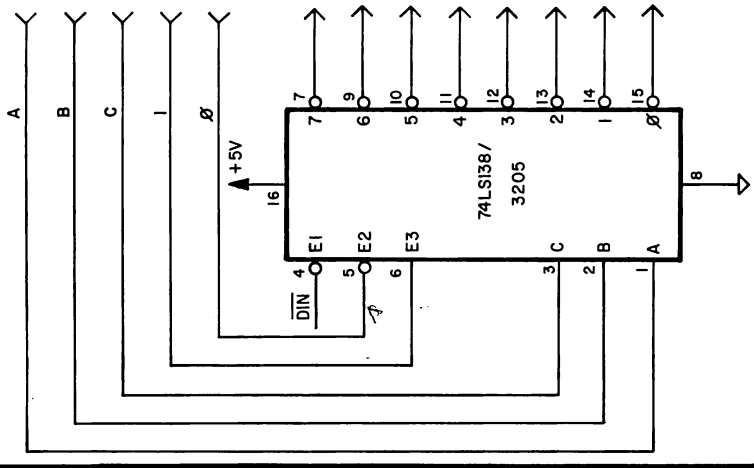
Note: The 74LS138 and 3205 are functionally identical, with identical packaging. Both are low-power Schottky devices operating at TTL voltage levels. However, the 74LS138 uses DTL-type input circuitry, while the 3205 uses TTL-type inputs. Effectively the 3205 has slightly

REV
A

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	DH1	DH1	DH2	DH3	DH4	DH5	DH1	DH2		$\overline{DT35}$	$\overline{DT33}$	$\overline{DT31}$	$\overline{DT25}$	$\overline{DT21}$	$\overline{DT15}$	$\overline{DT11}$	$\overline{DN5}$	$\overline{DN1}$		
	DH2	DH2	DH1	DH4	DH3	DH5	DH1	DH3		$\overline{DT33}$	$\overline{DT31}$	$\overline{DT23}$	$\overline{DT21}$	$\overline{DT13}$	$\overline{DT11}$	$\overline{DN3}$	$\overline{DN1}$			
	DH3	DH3	DH4	DH1	DH5	DH2	DH4	DH5		$\overline{DT33}$	$\overline{DT32}$	$\overline{DT23}$	$\overline{DT22}$	$\overline{DT13}$	$\overline{DT12}$	$\overline{DN3}$	$\overline{DN2}$			
	DH4	DH4	DH5	DH2	DH1	DH3	DH5	DH4		$\overline{DT27}$	$\overline{DT25}$	$\overline{DT23}$	$\overline{DT21}$	$\overline{DN7}$	$\overline{DN5}$	$\overline{DN3}$	$\overline{DN1}$			
	DH5	DH5	DH1	DH3	DH2	DH4	DH5			$\overline{DT27}$	$\overline{DT26}$	$\overline{DT23}$	$\overline{DT22}$	$\overline{DN7}$	$\overline{DN6}$	$\overline{DN3}$	$\overline{DN2}$			
	DH1	DH1	DH3	DH4	DH2	DH5	DH3			$\overline{DT27}$	$\overline{DT26}$	$\overline{DT25}$	$\overline{DT24}$	$\overline{DN7}$	$\overline{DN6}$	$\overline{DN5}$	$\overline{DN4}$			
	DH2	DH2	DH4	DH5	DH3	DH1	DH4			$\overline{DT17}$	$\overline{DT15}$	$\overline{DT13}$	$\overline{DT11}$	$\overline{DN7}$	$\overline{DN5}$	$\overline{DN3}$	$\overline{DN1}$			
	DH3	DH3	DH5	DH1	DH4	DH2	DH5			$\overline{DT17}$	$\overline{DT16}$	$\overline{DT13}$	$\overline{DT12}$	$\overline{DN7}$	$\overline{DN6}$	$\overline{DN3}$	$\overline{DN2}$			
	DH4	DH4	DH1	DH2	DH5	DH3				$\overline{DT17}$	$\overline{DT16}$	$\overline{DT15}$	$\overline{DT14}$	$\overline{DN7}$	$\overline{DN6}$	$\overline{DN5}$	$\overline{DN4}$			
	DH5	DH5	DH2	DH3	DH1	DH4				$\overline{DT17}$	$\overline{DT15}$	$\overline{DT14}$	$\overline{DT13}$	$\overline{DT12}$	$\overline{DT11}$	$\overline{DT10}$				
	DH1	DH1	DH3	DH4	DH5	DH2				$\overline{DT36}$	$\overline{DT32}$	$\overline{DT26}$	$\overline{DT22}$	$\overline{DT16}$	$\overline{DT12}$	$\overline{DN6}$	$\overline{DN2}$			
	DH2	DH2	DH4	DH5	DH1	DH3				$\overline{DT36}$	$\overline{DT34}$	$\overline{DT26}$	$\overline{DT24}$	$\overline{DT16}$	$\overline{DT14}$	$\overline{DN6}$	$\overline{DN4}$			
	DH3	DH3	DH5	DH1	DH2	DH4				$\overline{DT35}$	$\overline{DT34}$	$\overline{DT25}$	$\overline{DT24}$	$\overline{DT15}$	$\overline{DT14}$	$\overline{DN5}$	$\overline{DN4}$			
	DH4	DH4	DH1	DH2	DH3	DH5				$\overline{DT36}$	$\overline{DT34}$	$\overline{DT32}$	$\overline{DT30}$	$\overline{DT16}$	$\overline{DT14}$	$\overline{DT12}$	$\overline{DT10}$			
	DH5	DH5	DH2	DH3	DH4	DH1				$\overline{DT35}$	$\overline{DT34}$	$\overline{DT31}$	$\overline{DT30}$	$\overline{DT15}$	$\overline{DT14}$	$\overline{DT11}$	$\overline{DT10}$			
	DH1	DH1	DH3	DH4	DH5	DH2				$\overline{DT33}$	$\overline{DT32}$	$\overline{DT31}$	$\overline{DT30}$	$\overline{DT13}$	$\overline{DT12}$	$\overline{DT11}$	$\overline{DT10}$			
	DH2	DH2	DH3	DH4	DH5	DH1				$\overline{DT36}$	$\overline{DT34}$	$\overline{DT32}$	$\overline{DT30}$	$\overline{DT26}$	$\overline{DT24}$	$\overline{DT22}$	$\overline{DT20}$			
	DH1	DH1	DH2	DH3	DH4	DH5				$\overline{DT35}$	$\overline{DT34}$	$\overline{DT31}$	$\overline{DT30}$	$\overline{DT25}$	$\overline{DT24}$	$\overline{DT21}$	$\overline{DT20}$			
	DH2	DH2	DH4	DH5	DH1	DH3				$\overline{DT33}$	$\overline{DT32}$	$\overline{DT31}$	$\overline{DT30}$	$\overline{DT23}$	$\overline{DT22}$	$\overline{DT21}$	$\overline{DT20}$			
	DH3	DH3	DH5	DH1	DH2	DH4				$\overline{DT27}$	$\overline{DT26}$	$\overline{DT25}$	$\overline{DT24}$	$\overline{DT23}$	$\overline{DT22}$	$\overline{DT21}$	$\overline{DT20}$			
	DH4	DH4	DH1	DH2	DH3	DH5														
	DH5	DH5	DH2	DH3	DH4	DH1														

NOTE: PIN CONNECTIONS ON MODULAR MICRO BUS:

$\overline{DIN}$.....28	DH3.....13
DH5.....11	DH2.....14
DH4.....12	DH1.....15



	1	2	3
A			
B			
C			
D			

NOTE: PROPRIETARY INFORMATION. NOT TO BE DISCLOSED WITHOUT EXPRESS WRITTEN PERMISSION.	DRAWN BY Schneider	CHECKED BY KB	DRAWING NO. DS 9001 FIG 1.4.1
	DATE 11/29/78	DATE 12/1/78	

EF4A

SEC. 1 CIRCUIT DESCRIPTION (cont'd)

improved noise immunity. Though possibly a factor on crowded memory boards, the difference may usually be ignored in I/O applications.

The remainder of this data sheet describes a sample circuit for providing a *modular micro* system with one eight-bit parallel input port and one eight-bit parallel output port, using the 8008-compatible I/O address mode. A kit of parts is available as KT9001.

1.5 *Output Port 24* The schematic diagram for the KT9001 Universal I/O Kit shows a single eight-bit parallel output port, output 24. The strobe decoder has been connected using combination number 15 from Figure 1.4.1, producing signals for output ports 27, 26, 25, and 24, and for input ports 7, 6, 5, and 4.

Output port 24 consists of a single 74273 octal latch. Its eight input terminals are connected to the low-order address bus, DL7-DL0. When an OUTPUT 24 instruction is executed in an *8008 microcomputer,* $\overline{DT24}$ goes low at $\overline{DIN}$ time. When $\overline{DT24}$ ends, going high, the '273 is clocked and the output data is latched up in the output port. The 74273 normally retains its output data indefinitely, so long as power is supplied. The output terminals, which drive up to 10 TTL loads, can be used for any eight-bit parallel data function.

In an *8080 or 6800 microcomputer,* this port is addressed using the same hardware, but different programming. The microprocessor loads output data into the low-order address register, and the I/O port address into the high-order address register. (See Fig. 1.3.1 for the appropriate high-order page address.) A memory write instruction is executed. The output data is latched up in output port 24.

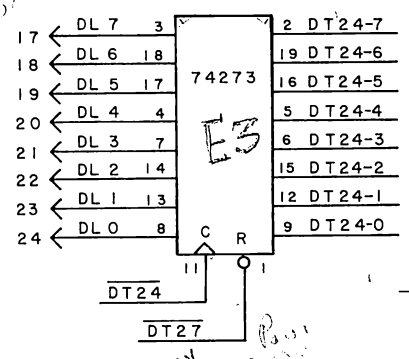
For illustration's sake, the 74273 reset (*R*) terminal is shown connected to $\overline{DT27}$. This latch can thus be initialized to an all-zero condition by causing the CPU to execute an OUTPUT 27 instruction. If resetting is unnecessary, this terminal can be connected through a protective resistor to +5 V. Ranging between 1K and 10K, a typical value is 3.3K.

1.6 *Input Port 4* The sample input port consists of two 8T97 bus drivers. When the $\overline{DN4}$ enable signal goes low, the 8T97 output terminals leave their inactive high-impedance state, and drive the microcomputer data bus with the eight-bit parallel input data present at the input terminals. A pair of 8T97s is used; each package includes six driver circuits, four with a common enable terminal (pin 1) and a pair with a separate shared enable terminal (pin 15). The two driver pairs enabled by pin 15 are unused in the sample circuit.

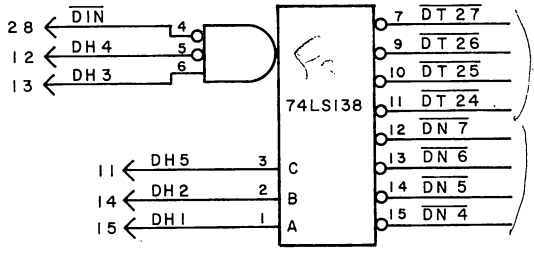
An 8008 microcomputer selects this input port by executing an INPUT 04 instruction. An 8080 or 6800 microcomputer selects this port by setting up the high-order address register to the appropriate page in memory (see Fig. 1.3.1) and executing a memory read instruction. (The low-order memory address need not be set up.) In all cases, the input data ends up in the accumulator.

REV
A

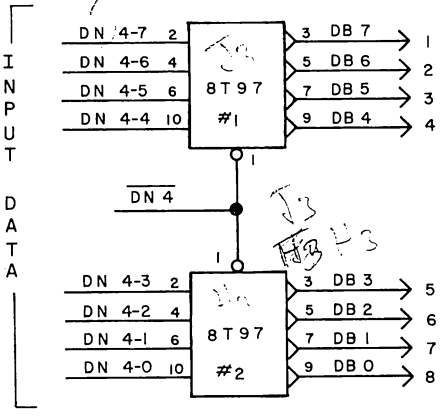
OUTPUT PORT Ø24



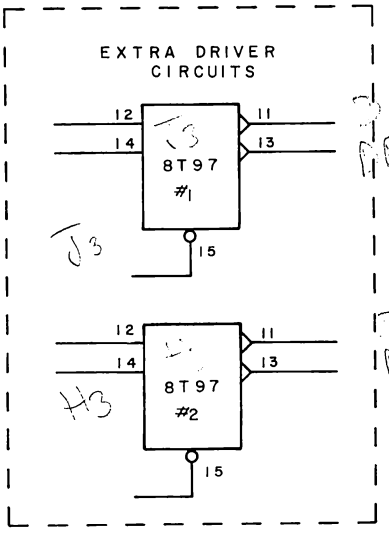
I/O STROBE DECODER



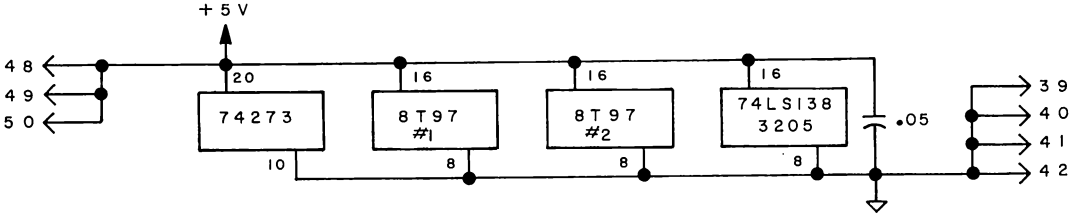
INPUT PORT ØØ4



EXTRA DRIVER CIRCUITS



POWER BUSSING



1	2	3
NOTE: PROPRIETARY INFORMATION. NOT TO BE DISCLOSED WITHOUT EXPRESS WRITTEN PERMISSION.	DRAWN BY Schneider DATE 11-19-75	CHECKED BY DH DATE 11-19-75	DRAWING NO. DS 9001-1 (Page 7)

EF4A

SEC. 1 CIRCUIT DESCRIPTION (cont'd)

1.7 *Combined Input/Output Instructions* The 8008 mode of I/O addressing offers a feature which is sometimes useful in microcomputer design: combined I/O instructions. Since output data is transmitted from the CPU via the low-order address bus (DL7-DL0), and input data is received by the CPU via the data bus (DB7-DB0), the same instruction can be used both to input data and to output data.

With the 8008, the combined I/O technique is limited to those eight instructions labeled as "input" instructions in the chart, Fig. 1.3.1. When one of these instructions, INP 00 through INP 07, is executed, the data in the 8008 A register travels to the DL address bus, just as it does during an output instruction. There it can be latched up by output port hardware, strobed by the input enable signal. The same input enable signal is also used to activate a three-state bus driver, transmitting input data to the CPU. During $\overline{\text{DIN}}$ time, data flows outwards from the CPU (via the DL bus), and inwards toward the CPU (via the data bus). In short, the 8008 "input" instructions are really *input/output* instructions.

With the 8080 or 6800, the combined I/O technique applies to all 32 I/O ports which are addressed as memory. A strobe/enable signal developed by the strobe decoder both strobes an output port, and enables an input port bus driver. At $\overline{\text{DIN}}$ time, data flow is bidirectional--moving via the DL bus (output data) and DB bus (input data). When the low-order memory address has previously been loaded with the output data, a memory read instruction causes the output data to end up in the output port, and input data to end up in the CPU.

SEC. 2 PARTS LIST

2.1 *Basic I/O Kit* The following parts are provided with KT9001-1.

Part No.	Quantity	Description
1053	3	SOCKET, 16-pin DIP, wirewrap, tin
1055	1	SOCKET, 20-pin DIP, wirewrap, tin
2042	1	IC, 74LS138/3205 3-to-8 decoder
2083	1	IC, 74273 octal latch
2132	2	IC, 8T97B hex bus driver
5501	1	CAPACITOR, .05 to 0.1 uf, 10 V min., ceramic

2.2 *Accessories* To expand I/O capabilities, the user can add additional bus drivers and latches. Since the single strobe decoder provided with the basic kit provides eight strobe/enable signals, additional drivers and latches can be purchased separately from our Replacement Parts List, using the above part numbers.

The following parts are currently in use in the series of *modular micros* from Martin Research. As a service to our customers, these parts may be ordered directly from Martin Research, for replacement purposes. Please note that the prices shown reflect the costs of inventory, handling, paperwork, and small-order processing.

An additional charge of $1.00 is added to any order for less than $25.00 of materials. Payment in advance is required, except for rated firms; shipping costs will be added to orders requiring billing. Outside the United States: add 10%; minimum order, $25.00; write for invoice in advance.

Please use our part numbers when ordering replacement parts.

PART NO.	DESCRIPTION	PRICE
700	BINDER, three-ring, with 12 tabbed separators, for Modular Micro Manual	$ 4.00
1000	CONNECTOR, PC board header; 50-pin, rt. angle solder leads	3.00
1001	CONNECTOR, 50-pin transition, clamps onto ribbon cable	6.00
1002	CONNECTOR, PC board header, 50-pin, rt. angle wirewrap pins	7.50
1007	CONNECTOR, 20-pin transition, clamps onto ribbon cable	5.00
1008	CONNECTOR, PC board header, 20-pin, rt. angle wirewrap pins	4.00
1012	SOCKET, 14-pin DIP, solder-tail, tin--pack of 4	2.00
1013	SOCKET, 16-pin DIP, solder-tail, tin--pack of 4	2.00
1014	SOCKET, 18-pin DIP, solder-tail, tin--pack of 2	1.80
1015	SOCKET, 20-pin DIP, solder-tail, tin--pack of 2	2.05
1017	SOCKET, 24 DIP, solder-tail, tin--pack of 2	1.40
1037	SOCKET, 24-pin DIP, solder-tail, gold--pack of 2	1.50
1052	SOCKET, 14-pin DIP, wirewrap, tin	1.50
1053	SOCKET, 16-pin DIP, wirewrap, tin	1.65
1055	SOCKET, 20-pin DIP, wirewrap, tin	2.40
1057	SOCKET, 24-pin DIP, wirewrap, tin	2.10
1059	SOCKET, 40-pin DIP, wirewrap, tin	3.60
1090	CONNECTOR, .062" male contact, kit of 20	.75
1501-XXX	CABLE, 20-conductor ribbon--XXX = length, in.; in 6-in. increments, 18 in. min.	.10/in.
1503-XXX	CABLE, 50-conductor ribbon--XXX = length, in.; in 6-in. increments, 18 in. min.	.15/in.
2000	IC, 7400 quad NAND gate--pack of 2	1.00
2004	IC, 7404 hex inverter--pack of 2	1.00
2005	IC, 74LS04 hex inverter--pack of 2	1.40
2007	IC, 74LS08 quad AND gate--pack of 2	1.30
2008	IC, 7410 triple 3-input NAND--pack of 2	1.00
2009	IC, 74LS10, triple 3-input NAND--pack of 2	1.30
2010	IC, 7413 dual 4-input NAND, Schmitt trigger	1.20
2012	IC, 7416 hex inverter/OC buffer--pack of 2	1.30
2013	IC, 7417 hex OC buffer--pack of 2	1.30

REPLACEMENT PARTS LIST

PART NO.	DESCRIPTION	PRICE
2014	IC, 7420 dual 4-input NAND--pack of 2	$ 1.00
2016	IC, 7427 8-input NAND--pack of 2	1.00
2029	IC, 7474 dual D flip-flop--pack of 2	1.40
2030	IC, 74LS74 dual D flip-flop--pack of 2	1.70
2034	IC, 7490 4-bit BCD counter	1.00
2035	IC, 7493 4-bit binary counter	1.00
2037	IC, 74122 one-shot	1.00
2041	IC, 74125 hex tri-state driver	1.05
2042	IC, 74LS138 3-to-8 decoder	2.10
2043	IC, 74LS139 dual 2-4 decoder	1.85
2044	IC, 74150 16-to-1 multiplexer	3.30
2050	IC, 74154/9311 4-to-16 decoder	2.20
2083	IC, 74273 octal latch	3.00
2105	IC, 74173 4-bit latch, tri-state	3.00
2107	IC, 74LS174 hex D flip-flop	3.50
2130	IC, 8T28B quad bi-bus driver	3.70
2132	IC, 8T97B hex bus driver	2.05
2133	IC, 8216 quad bi-bus driver	10.00
2140	IC, 8859, hex LED driver	3.60
2150	IC, 9368 hex LED digit driver	2.25
2200	IC, 1702A-6 256 x 8 PROM (unprog.)	25.00
2202	IC, 8008 central processing unit	40.00
2203	IC, 8008-1, premium CPU	52.00
2204	IC, 8080 central processing unit	60.00
2205	IC, 2112 256 x 4 RAM, 1.0 us	4.75
2207	IC, 2102A-4 1024 x 1 RAM, 450 ns	4.50
2401	IC, 4001 quad NOR gate, CMOS--pack of 2	1.40
2420	IC, 4050 hex buffer, CMOS	1.50
3000	TRANSISTOR, 2N5138 PNP--pack of 3	1.00
3050	TRANSISTOR, 2N3019--pack of 2	1.00
3500	LED, red, with red frosted lens--pack of 6	1.00
3501	LED, red, with clear lens--pack of 6	1.00
3502	LED, MAN-74 7-segment digit	3.00
3700-05	DIODE, IN5225 zener, 3.0 V, 5%, 500 mw	.75
3702-01	DIODE, SD1, 1A, 50 V--pack of 3	1.00
4000-25	RESISTOR, 10 ohms, 10%, ¼w--pack of 10	1.00
4000-27	RESISTOR, 15 ohms, 10%, ¼w--pack of 10	1.00
4000-42	RESISTOR, 270 ohms, 10%, ¼w--pack of 10	1.00
4000-43	RESISTOR, 330 ohms, 10%, ¼w--pack of 10	1.00
4000-44	RESISTOR, 390 ohms, 10%, ¼w--pack of 10	1.00
4000-49	RESISTOR, 1K, 10%, ¼w--pack of 10	1.00
4000-55	RESISTOR, 3.3K, 10%, ¼w--pack of 10	1.00

PART NO.	DESCRIPTION	PRICE
4000-61	RESISTOR, 10K, 10%, ¼w--pack of 10	$ 1.00
4000-66	RESISTOR, 27K, 10%, ¼w--pack of 10	1.00
4000-73	RESISTOR, 100K, 10%, ¼w--pack of 10	1.00
4001-97	RESISTOR, 1K, 5%, ¼w--pack of 5	1.00
4001-105	RESISTOR, 2.2K, 5%, ¼w--pack of 5	1.00
4001-115	RESISTOR, 5.6K, 5%, ¼w--pack of 5	1.00
4001-122	RESISTOR, 11K, 5%, ¼w--pack of 5	1.00
5000-16	CAPACITOR, 220 pf, 1000 V, ceramic--pack of 2	1.00
5010	CAPACITOR, 0.68 uf, 20%, 15 V, tantalum	.60
5011	CAPACITOR, 100 uf, 25 V, electrolytic	1.00
5501	CAPACITOR, .05 to 0.1 uf, 10 V min., ceramic--pack of 6	1.00
5502	CAPACITOR, 22 to 50 uf, 15 V min., tantalum--pack of 2	1.00
5900-1	CRYSTAL, 12.8 MHz	9.60
5900-2	CRYSTAL, 16.0 MHz	6.45
6000	HARDWARE, 5/8" hex spacer, M & F ends, 6-32 threads--pack of 16	2.00
6050-2	HARDWARE, 6-32 binder-head machine screw, ¼"--pack of 12	1.00
6500	KEYBOARD, 20 keys, calculator style, on PC board	6.00

October, 1975

martin research

3336 commercial avenue / northbrook, illinois 60062 / (312) 498–5060

WARRANTY POLICY

Martin Research, Manufacturer, warrants that the products delivered to the purchaser are free from defects in material and workmanship under normal use and service. Manufacturer's obligation under this warranty is limited to replacing or repairing or giving credit for, at its option, at its factory, any product which shall, within one (1) year after date of invoice, be returned to the Manufacturer's factory of origin, transportation charges prepaid, and which are, after examination, disclosed to the Manufacturer's satisfaction to be thus defective. This Warranty is expressed in lieu of all other warranties, expressed, statutory, or implied, including the implied warranty of fitness for a particular purpose, and of all other obligations or liabilities on the Manufacturer's part, and it neither assumes nor authorizes any other person to assume for the Manufacturer any other liabilities in connection with the sale of said articles.

This Warranty shall not apply to any of such products which shall have been repaired or altered, except by the Manufacturer, or which shall have been subjected to misuse, negligence, or accident. The aforementioned provisions do not extend the original warranty periods of any product which has either been repaired or replaced by the Manufacturer.

RETURN POLICY

I. *AUTHORIZATION REQUIRED*

Before returning any equipment to the Manufacturer for repair, replacement, or refund, authorization must first be received from Martin Research. Written authorization is normally required.

II. *SHIPPING, PACKING, AND INSURANCE*

Equipment returned to the Manufacturer should be securely packed in a rigid container with ample cushioning material. Due precautions must be taken with electrically sensitive devices, such as MOS (metal oxide semiconductor) circuitry, including packing in conductive material. *Equipment should never be shipped unless it is insured for its full value.* In case of damage, the party shipping the equipment will be notified promptly. As a matter of law, and of policy on the part of shipping agencies, the responsibility for filing damage claims rests with the shipping party, though Martin Research will verify any such claim at the shipper's request. Since payment by shipping agencies on claims of damage may not be prompt, we urge our customers to take extra care in packaging equipment for shipment.

(Continued)

Return Policy - Cont'd.

III. *DAMAGED AND OUT-OF-WARRANTY EQUIPMENT*

If it appears to the Manufacturer that returned equipment has been subjected to misuse, negligence, or accident by the user, the user shall be notified promptly. The Manufacturer retains the option to return such equipment unrepaired, and to charge for shipping in such event. If the user has so authorized, the Manufacturer will repair such equipment, or equipment whose warranty period has expired, for any replacement parts deemed necessary at the rates shown on its current Replacement Parts List, and charging for service at the current hourly rate for electronic repair. Estimates will be provided upon request.

REV. A

Date: November 25, 1975

Applicability: (1) All 423 PROM/RAM boards equipped with 8T28 bus drivers;

(2) Model 423 PROM/RAM boards marked *423A.*

Action Required: This Service Note lists a restriction on the physical positioning on the *modular micro* bus, in relationship to the CPU board, of Model 423 boards which are equipped with type 8T28 bus drivers. The distance between any such Model 423 and the CPU board should not exceed two inches (5 cm). Details appear in Section 1.

This Note also calls for certain modifications to be made to the 423 board, *A* version, to bring its circuitry into line with the 423 *B* schematic. These changes, necessary for proper operation of the 423 board, are described in Section 2.

SEC. 1 POSITIONING OF 423 BOARDS EQUIPPED WITH 8T28 BUS DRIVERS

The type 8T28 bus driver has proven in field testing to have a defect in operation which adversely affects the 423 PROM/RAM board. The problem occurs when a PROM or RAM on the 423 board presents the input terminals of the 8T28 with a logic high value (near +5 volts), and the 8T28 is then activated at pin 15 by the READ signal, to transmit data to the CPU via the data bus. The 8T28 output should leave its high-impedance state and move directly to a *high* voltage level. Instead, 8T28 devices currently being supplied always go momentarily to a logic *low* level. Instead, 8T28 devices currently being supplied always go momentarily to a logic *low* level, then assume the proper high state. The result is a 10 ns negative-going glitch on each data bus line driven with a logic one input. When all eight lines are *ones*--that is, when the value being read from memory is octal 377--the resultant high-frequency, high-current pulse is sufficient to cause oscillations on the 423 board. The microcomputer reads the value *000* instead of the *377* stored. (Sometimes this happens with stored words consisting of only seven *1s,* such as 376, 277, etc.)

This effect does not occur when the distance between the CPU board and the 423 PROM/RAM board is minimal. For this reason, the following restriction is placed on the position of the 423 board:

THE DISTANCE BETWEEN A MODEL 423 PROM/RAM BOARD EQUIPPED WITH 8T28 BUS DRIVERS, AND THE CPU BOARD IN THE SYSTEM, SHALL NOT EXCEED TWO INCHES (FIVE CENTIMETERS).

This distance is measured along the length of the fifty-wire flexible cable between the respective cable connectors for the CPU and PROM/RAM boards.

Modular micro systems based on the 8080 and 6800 use type 8T26 inverting bus drivers on the 423 board, which are pin-compatible with the 8T28, but do not exhibit this problem. These devices are provided with the 8080 Update Kit, for upgrading on 8008 system, or may be purchased from our Replacement Parts List. The position limitation on the 423 does not apply when

SEC. 1 POSITIONING OF 423 BOARDS EQUIPPED WITH 8T28 BUS DRIVERS (cont'd)

8T26s are used. However, note that inverting bus data usually affects the software used in the system.

When contacted by Martin Research, a spokesman for the 8T28 manufacturer, Signetics, stated that the problem lies in the output stage. The chip is being redesigned. Martin Research will notify *modular micro* Manual holders when a redesigned 8T28 becomes available to us. The corrected IC would remove the positioning constraint on the 423 board.

SEC. 2 CORRECTIONS TO 423 PROM/RAM BOARD, *A* VERSION

Model 423 PROM/RAM boards marked *423A*--on the tinned copper foil or in the white silk-screened legend--must be modified for proper operation. The schematic in the 423 data sheet, *B* revision, shows the proper way of connecting the 423 board. For information only, page one *only* of the 423A schematic is reprinted with this Service Note.

The correction is necessary to prevent the PROM/RAM board from erroneously destroying data stored in RAM one location *beyond* the location currently addressed by the CPU.

The following changes to the 423A are necessary:

(A) Add a 74LS139 at location F3. Shown on the 423A schematic as "optional," the previously-unused second half of this chip is necessary in the revised circuit. If this second 74LS139 is missing, contact Martin Research.

(B) Make two cuts on the board, solder side, as indicated by the two *X*'s in Figure 2.1.1. A knife or other sharp edge may be used. Be sure that the foil connection is completely broken.

(C) Make five connections, as shown by the heavy black lines on Fig. 2.1. Use insulated hookup wire, stripping very short bare ends, and soldering to the IC terminals shown. The routing of the wires in the pictorial is indirect, for the sake of clarity. However, wire routing should be direct as possible, with just enough extra length to permit shaping the ends, preventing shorting to adjacent IC terminals. The five wires are connected as follows:

(1) After making the cut marked with the heavy *X*, connect a short length of bare wire between pin 14 of IC F1 to the vertical trace which has just been cut. It will be necessary first to scrape this vertical trace to remove the green solder mask. Make sure not to short against the adjacent vertical trace.

1975 Martin Research, Northbrook, IL 60062, USA. All rights reserved.

4	3	2	Rev.	ECO	By	Ck	Date
A	A	A	A	016	BR	DM	12.5.75

REV. A

SEC. 2 CORRECTIONS TO 423 PROM/RAM BOARD, *A* VERSION (cont'd)

(2) Connect a wire from pin 36 on the 50-pin bus header, to pin 14 at IC F3. This wire is #2 on the pictorial.

(3) Having made a cut as indicated by the light *X*, near IC E3 on the board, connect the left-hand terminal on this short island to pin 12 of IC F3 (wire #3).

(4) Now connect a wire from the right-hand hole on the E3 island, to pin 13 of IC F3 (wire #3).

(5) Now connect a short wire (#5) between pins 15 and 8 of IC F3.

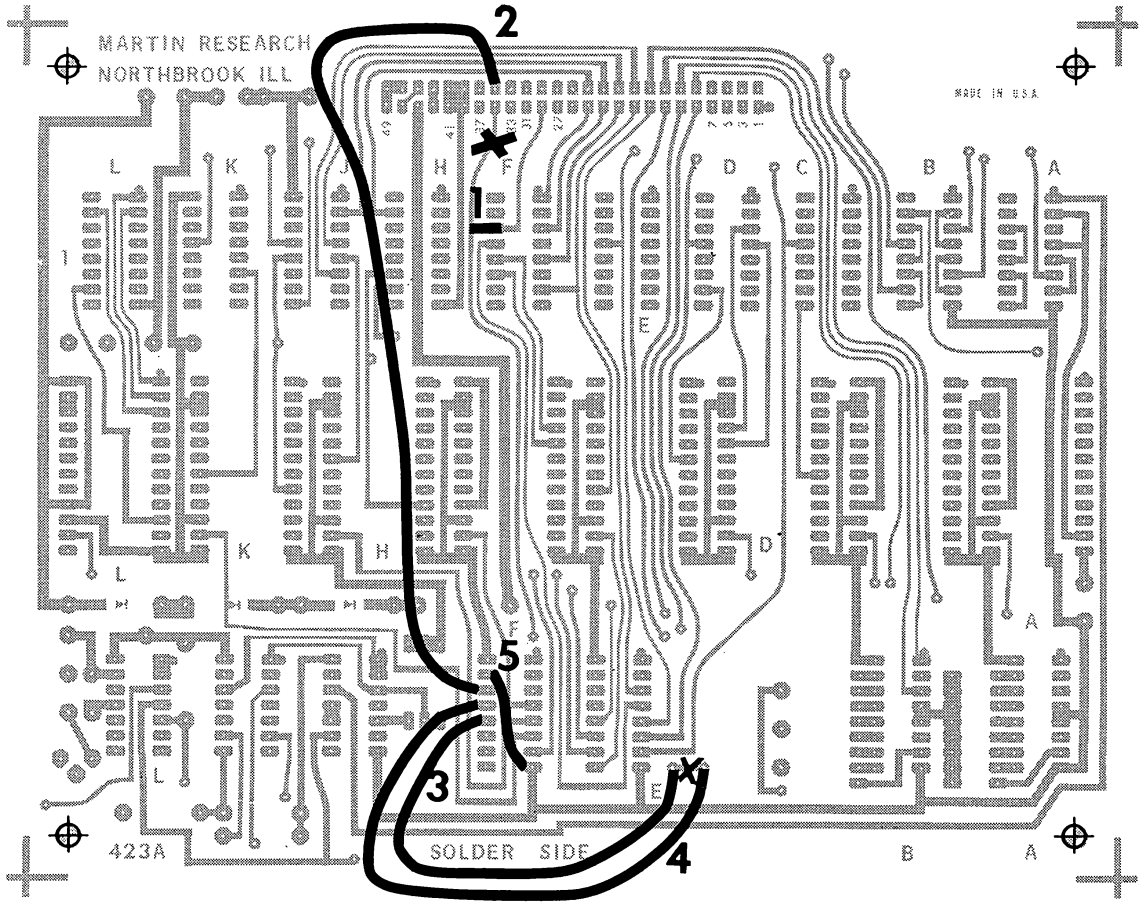
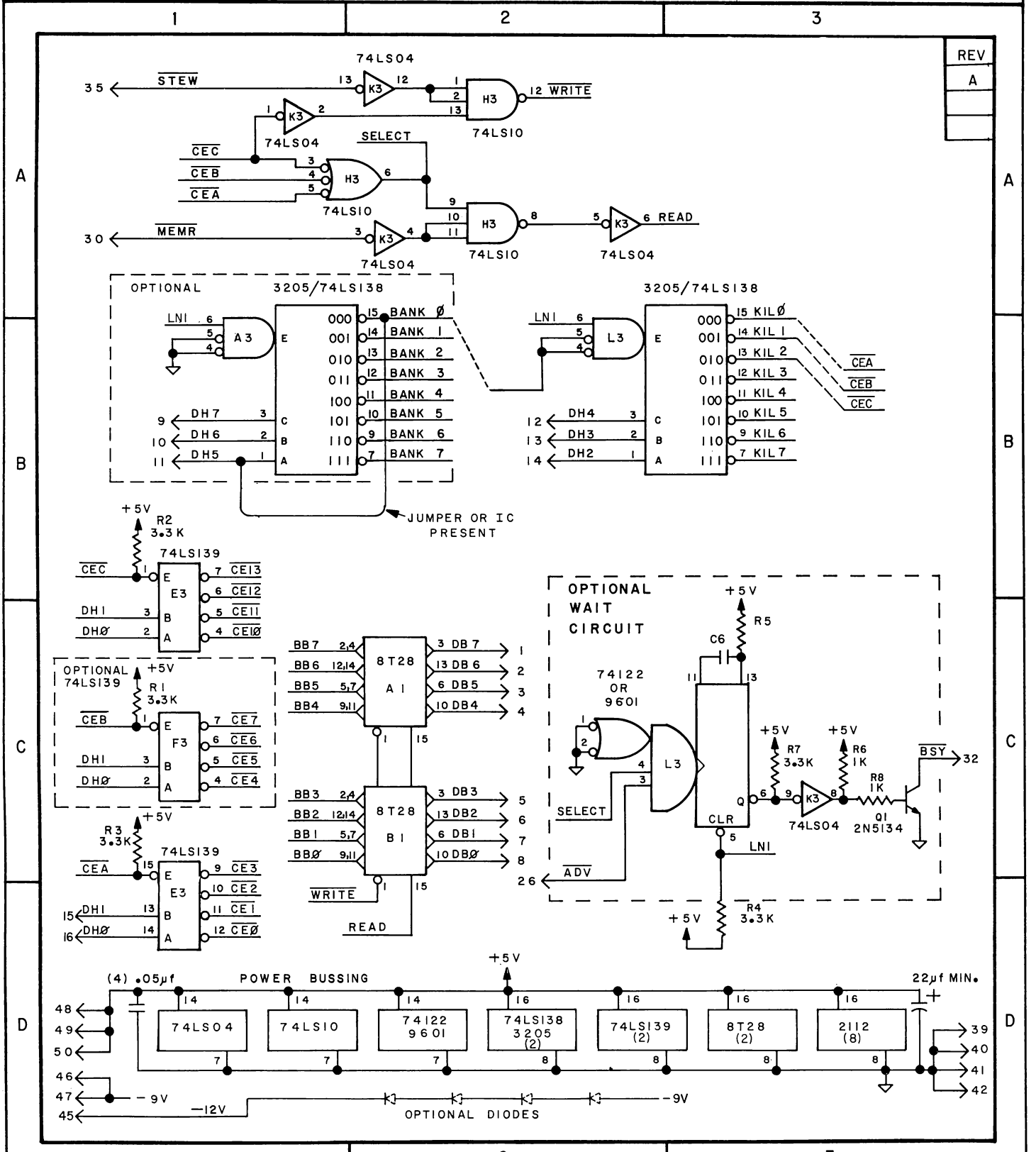


Figure 2.1--Modifications to 423A PROM/RAM Board

If the 423 board fails to operate correctly after these changes have been made, recheck the wiring carefully. If the board still fails, contact Martin Research for assistance.

423A boards shipped from our factory after the date of this Service Note have already been modified to correspond with the above changes.



NOTE: PROPRIETARY INFORMATION. NOT TO BE DISCLOSED WITHOUT EXPRESS WRITTEN PERMISSION.	DRAWN BY Roeck DATE 11-20-75	CHECKED BY DATE 12/1/75
		DRAWING NO. SV 423-1 (Page 4)

EF4A

modular micros
martin research

REV. A

REGISTER INSTRUCTIONS		DESTINATION							
EX.: LAM= LOAD ACCUMULATOR FROM MEMORY		A	B	C	D	E	H	L	M
A=0 D=3 L=6 B=1 E=4 M=7 C=2 H=5	SOURCE A	300	310	320	330	340	350	360	370
	SOURCE B	301	311	321	331	341	351	361	371
	SOURCE C	302	312	322	332	342	352	362	372
	SOURCE D	303	313	323	333	343	353	363	373
	SOURCE E	304	314	324	334	344	354	364	374
	SOURCE H	305	315	325	335	345	355	365	375
	SOURCE L	306	316	326	336	346	356	366	376
	SOURCE M	307	317	327	337	347	357	367	—
	MNE-MONIC	A	B	C	D	E	H	L	M
LOAD REGISTER IMMEDIATE	Lrl	006	016	026	036	046	056	066	076
INCREMENT REGISTER	INr	—	010	020	030	040	050	060	—
DECREMENT REGISTER	DCr	—	011	021	031	041	051	061	—

ALU INSTRUCTIONS	MNE-MONIC	A	B	C	D	E	H	L	M	IMM.
ADDr TO A→ A	ADr	200	201	202	203	204	205	206	207	004
ADD r + C TO A→ A	ACr	210	211	212	213	214	215	216	217	014
SUB. r FROM A→ A	SUr	220	221	222	223	224	225	226	227	024
SUB. (r + C) FROM A→ A	SBr	230	231	232	233	234	235	236	237	034
AND r WITH A→ A	NDr	240	241	242	243	244	245	246	247	044
EXCL. OR r WITH A→ A	XRr	250	251	252	253	254	255	256	257	054
OR r WITH A→ A	ORr	260	261	262	263	264	265	266	267	064
COMPARE r WITH A	CPr	270	271	272	273	274	275	276	277	074

		CODE
ROTATE A LEFT	RLC	002
ROTATE A RIGHT	RRC	012
ROTATE A LEFT THRU CARRY	RAL	022
ROTATE A RIGHT THRU CARRY	RAR	032

FLAGS AFFECTED	C	Z	S	P
INCREMENT/DECREMENT	—	✓	✓	✓
ALU INSTRUCTIONS	✓	✓	✓	✓
ROTATE	✓	—	—	—

PC/STACK INSTRUCTIONS												
	UNCONDITIONAL		FALSE CONDITION					TRUE CONDITION				
OPERATION	MNE-MONIC	*	MNE-MONIC	CARRY	ZERO	SIGN	PARITY	MNE-MONIC	CARRY	ZERO	SIGN	PARITY
JUMP	JMP	1x4	JFc	100	110	120	130	JTc	140	150	160	170
CALL	CAL	1x6	CFc	102	112	122	132	CFc	142	152	162	172
RETURN	RET	0x7	RFc	103	113	123	133	RFc	143	153	163	173
	MNEMONIC		CODE									
RESTART	RST OaO		Oa5									

* X = DON'T CARE (0-7)

I/O INSTR.	INP n	INP n	OUT 1n	OUT 2n	OUT 3n
	0	101	121	141	161
	1	103	123	143	163
	2	105	125	145	165
	3	107	127	147	167
	4	111	131	151	171
	5	113	133	153	173
	6	115	135	155	175
	7	117	137	157	177

UNDEFINED INSTRUCTION CODES
042
052
062
070
071
072

HALT	HLT	000
	HLT	001
	HLT	377

modular micros martin research

The MONITOR 8 is an extensive software package allowing the user complete facility in programming an 8008 microcomputer. The program interfaces a 110-baud serial in, serial out ASCII teleprinter--such as the Teletype® Model 33, KSR or ASR; and a *MIKE 2* microcomputer. MONITOR 8 is available in the form of seven type 1702A-6 PROMs, as Martin Research part number KT2210-1.

The MONITOR 8 supports a number of useful functions, including an Octal Dump (print a listing of data stored in memory at selected addresses); Dump Symbolic (print a listing of instructions stored in memory, including mnemonics for the 8008 instruction set); an Edit routine; Set Breakpoint (for software debugging); and others, as explained below. In short, MONITOR 8 is a complete software operating system for the *MIKE 2* microcomputer.

The MONITOR 8 requires the following equipment: a *MIKE 2* microcomputer, with a Model 403 or 423 PROM/RAM board, including page 013 in RAM and sockets for seven PROMs; a Model 33 Teletype®, or equivalent, and suitable hardware interfacing circuitry--such as Martin Research KT9004-1; and the seven MONITOR 8 PROMs, KT2210-1.

Modified versions of the MONITOR 8, suitable for use with other types and speeds of printer, are also available from Martin Research.

TECHNICAL DATA

SEC. 1 PROGRAM DESCRIPTION

1.1 *System Start-Up* Ensure power is off to the PROM programmer, if one is provided. TTY should be set to local. Apply power to the microcomputer. Turn the TTY to on line and interrupt the microcomputer. This may be done either by using the Console Board interrupt keys on the MIKE 2 Console Board, or with the switch provided with the ASCII keyboard interface hardware. The TTY should respond with a carriage return and a line feed (CRLF), should print eight dashes, and should perform another CRLF.

1.2 *Addressing* The memory in the *MIKE 2* is organized into banks. Each bank is 0 to 377 octal (256 decimal) bytes in length. When communicating with MONITOR 8 the addresses take the following form:

N_5 N_4 N_3 N_2 N_1 N_0

$N_0 - N_5$ are octal digits with the following significance:

N_5 = Special modifier value 0-3 possible.

N_5 = 0 or 1 memory accessed is normal ROM or RAM.

N_5 = 2 or 3 memory accessed is the PROM in the programming station, if one is attached to the system.

SEC. 1 PROGRAM DESCRIPTION (cont'd)

N_4 N_3 = Memory page number, value 00-77 possible

N_4 N_3 = 00 to 07 memory accessed is PROM. N_4N_3 = 10 to 13 in RAM.

N_2 N_1 N_0 = Byte location within page, value 000 to 377 possible.

1.3 MONITOR 8 COMMAND SUMMARY

The MONITOR 8 system is now ready to load symbolic program input or accept one of the following utility commands.

LOC	(Set current location pointer)	EDT	(Enter edit mode)
DLP	(Display current location pointer)	XQT	(Initiate program execution)
DPS	(Dump symbolic)	CPY	(Copy routine)
LDO	(Load octal)	TRN	(Translate routine)
DPO	(Dump octal)	SBP	(Set break-point)
LBF	(Load BNPF format)	CBP	(Clear break-point)
DBF	(Dump BNPF format)	PRG	(Program PROM)

1.4 LOC (Set Current Location Pointer)

All data entry and manipulation is done at the address indicated by the current location pointer (CLP). The pointer value is stored and used by the MONITOR software. After each machine instruction is entered the CLP is updated to point at the next available memory location. The two pseudo-operators LOC and DLP allow the user to preset and display the current location pointer.

When LOC is typed the machine responds with a space (b). The user must then specify a six-digit address (see addressing). After the last address digit has been entered, the machine responds with CRLF and waits for the next command. The MONITOR software uses RAM addresses 013350-013377 inclusive, but all other addresses are available to the user.

1.5 DLP (Display Current Location Pointer)

If the user wishes to display the CLP, he may type in DLP. The machine responds by typing out the CLP and then performs a CRLF and waits for the next instruction.

1975 Martin Research, Northbrook, IL 60062 USA. All rights reserved.

22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	Rev.	ECO	By	Ck	Date
A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	16	BR	DM	12/4/75

SEC. 1 PROGRAM DESCRIPTION (cont'd)

NOTE: The CLP is destroyed by several of the Monitor routines. When this is the case, the Monitor will print 8 dashes on completion of the requested function. In these instances, the user should respecify the CLP using the LOC command before proceeding.

1.6 SYMBOLIC PROGRAM INPUT

Once the CLP has been initialized, the user may type in his program. After each mnemonic instruction has been entered, the machine will respond with a CRLF or, if the instruction requires an argument, with a space. All immediate instructions require a 3 digit octal data byte. All jump and call commands require a 6 digit split octal address (see Addressing). Input/output and restart instructions require a 3 digit octal number to specify a port number or restart address. After the instruction and the corresponding argument have been entered, a CRLF will be generated and the next instruction may then be entered. After each entry, the CLP is automatically updated to point to the next available memory location.

There are several bit combinations which will be interpreted by the 8008 as a halt command. The following commands will be interpreted by the MONITOR as HALT command bit combinations.

Mnemonic	Resultant Octal	8008 Interpretation
HLT	000	HLT
INA	000	HLT
DCA	001	HLT
LMM	377	HLT

1.7 DPS (DUMP SYMBOLIC)

A symbolic listing is generated by typing DPS. The machine will respond with a CRLF and a * (this is the prompter indicating that the machine requires further address information). The user must now type in the initial and final address, defining the block of code to be dumped. These two addresses must be entered as a 6-digit split octal number (see addressing). When the initial address has been entered, the machine responds with a blank and awaits the final address. Then the final address has been entered the machine responds with 3 CRLF's and commences listing. The listing includes the current memory address, the octal instruction and the mnemonic. For a multi-byte-instruction the listed address is that of the first byte of the instruction. Any data fields associated with the instruction (immediate data, addresses, I/O port numbers or restart addresses) will be printed following the mnemonic. One instruction is listed per line with 62 lines generated per page. An auto paging feature separates each 11" page by 3 CRLF's. Invalid instructions are displayed as ? ? ?.

SEC. 1 PROGRAM DESCRIPTION (cont'd)

1.8 LDO (Load Octal)

Typing LDO will initiate the octal load routine. As in the dump routine, the machine waits for two octal addresses. It then outputs a CRLF and will begin reading in from the keyboard or tape reader. Each line which contains data must have a / symbol to the left of the data field. Each 3 digit octal value which follows the / is interpreted as data. Leading zeros must be included and each value must be separated by at least 1 blank. Any data to the left of the first / is ignored (note that this is usually the address generated by the DPO routine). When the final address specified has been filled, the routine returns to the MONITOR.

1.9 DPO (Dump Octal)

The dump routine will list 8 three-digit octal values per line. Each line is started by the current address followed by a /. The user must specify the starting and ending address. When a DPO is typed the machine will respond with a CRLF *. The first valid octal digit (0-7) typed will be interpreted as the beginning of the "initial address" (see Addressing). After N_0 has been entered the machine will respond with a space. Next the ending address must be typed. After both addresses are entered the machine does a CRLF and starts the dump. It will continue until it has typed the final location, and then return to the controller.

If a dump of the PROM station is required the address is specified by a 2 or 3 in the modifier bits of the address (see Addressing).

1.10 LBF (Load BNPF Format)

The BNPF load routine is similar to the octal load routine in its initiation. The initial and final addresses are entered and any key will initiate the load. A B signifies the start of a data field and F signifies the end. All enclosed characters must be either P's (1) or N's (0). If a format error occurs, the present memory location is displayed followed by a ? and control returns to the MONITOR.

1.11 DBF (Dump BNPF Format)

If a DBF command is run the machine responds with a CRLF and waits for a starting and ending address. These must have the same format as in an octal dump. After the final address is entered a CRLF is typed and the machine starts the dump. Each memory location is listed sequentially, five bytes to a line. The dump or BNPF dump routines as described in this manual will list any portion of memory, including the PROM programming station (if one is attached to the system).

1.12 EDT (Enter EDIT Mode)

The edit mode is entered by typing EDT. The editor responds with a CRLF and types the

SEC. 1 PROGRAM DESCRIPTION (cont'd)

value of the CLP followed by a /. It is now ready to accept one of the following commands:

nnn	— Where nnn is a three-digit octal value to be loaded into memory.
b	— Display memory value
	— Decrement the current location pointer
*AAAAAA	— Redefine the current location pointer with the value AAAAAA
@	— Equivalent to XQT
R	— Return to the MONITOR

If data is to be loaded it must immediately follow the / symbol. An invalid symbol will cause a CRLF with the CLP retyped. The nnn value is assembled as an 8-bit word and stored in the memory. Attempting to write into a ROM address will not be flagged, yet the data will not (cannot) be written.

If a blank is entered after the / the contents of the current memory location will be displayed. Two options are then available:

a) — nnn Replace the current value with nnn.

b) Any other symbol will increment the CLP.

Following the CLP / the editor examines the first character inputted to determine the command. If data is to be inputted immediately, it must be in the first three locations following the /. If the data follows a — (used to replace displayed data) the input is relatively format-free. The first octal digit will define the replacement data, any other symbols may appear between the — symbol and the data. The same is true of the *AAAAAA command. Following the command or data the editor types the new CLP on the next line and is ready to accept the next command.

1.13 XQT (Initiate Program Execution)

The XQT command allows the user to start the execution of his program. Following the typing of XQT the machine will respond with a space and wait for the starting address of the program. The entire user routine is treated as a subroutine which is called from the MONITOR. The user may return to MONITOR by including a RET (return) at the end of his routine.

1.14 CPY (Copy Routine)

Typing CPY will initiate a copy of blocks of memory. Like the dump and load routines, this routine requires a start address and an end address (defining the block to be moved). In addition, after the block end address has been entered, the machine will respond with a CRLF* and wait for the entry of a third address, the new start address for the block to be copied. After the third address has been entered, the entire block specified will be copied unchanged starting at the new

SEC. 1 PROGRAM DESCRIPTION (cont'd)

start address. When the copy has been completed control returns to the MONITOR.

1.15 TRN (Translate Routine)

Typing TRN when in the MONITOR mode initiates the translate. This routine is intended for use after a program is running in RAM and it is desired to store it in PROM which will reside in a different bank. No movement of data occurs, but all jump and call addresses which are internal to the bank will be changed to reflect the new specified bank. This routine again requires a start of block and an end of block address, to define the block to be operated on. After the second address has been entered, the machine responds with a CRLF. The machine is now waiting for two three-digit octal bank numbers (possible range 000 to 077). After the first bank number has been entered (the source bank number), the machine responds with a — and waits for the second bank number (the destination bank number). After the second bank number has been entered, the machine searches the specified block for all call and jump references to the source bank and changes these to refer to the destination bank. When the changes have been completed, the machine returns to the MONITOR mode.

1.16 SBP (Set Break Point)

Break-points allow the tracing of program flow during its execution. If a RST 060 command is encountered during program execution the MONITOR software will print out the contents of the carry flag, A B C L and H registers, and the memory contents addressed by the H and L registers, and then return to the monitor software.

The SBP command inserts a RST 060 command at the address specified by the user. The address at which the break-point is inserted and the instruction originally found there is retained by the MONITOR. Before setting subsequent break-points, the MONITOR will first restore the data at the previous break-point location.

1.17 CBP (Clear Break Point)

The CBP command will restore the data at the present break-point location.

1.18 PRG (Program PROM)

The MONITOR software also contains the facility for controlling a PROM programming station if one is attached to the system. The programming routine is entered by typing PRG. The programming routine will allow programming a PROM with data presently located in memory. An initial and final address must be specified. The routine will program the data from specified location to the corresponding word location within the PROM.

SEC. 1 PROGRAM DESCRIPTION (cont'd)

E.G. 010177 Location 177 of the PROM

There is a one-to-one correspondence between the address being read within a bank and the address being programmed in the PROM.

To accommodate 1702A devices it is necessary to change the programming duty cycle from 2% to 20%. After receiving the initial and final address to be programmed the programming routine will respond with CRLF "%". The user must then type an "A" or "N" to determine the timing loop. Typing A will give a program pulse duty cycle of 20% for 1702A types and an "N" gives approximately 20% duty cycle for 1702 devices. There is no check for validity of the constants entered and under no condition should standard devices be programmed with excessive duty cycles.

The programming routine will first check if the PROM data is equal to the program data. If the byte patterns are identical the routine proceeds to the next address. If the location must be programmed the device is hit with a single program pulse and the data is again checked against the desired data. When the data is finally read as being valid, after B program pulses, the device is hit with an additional 4 x B program pulses.

It should be noted that attempting to program a standard device using 1702A timing (20% duty cycle) will destroy the device. Note also that the unprogrammed state for an A series device is all lows, whereas for standard devices it is all highs.

1.19 CONTROL A

Included in the TTY input routine is a check for the CTRLA key. Depressing the CTRL button and A key simultaneously will cause the machine to immediately return to the MONITOR routine, and is equivalent to a MONITOR restart.

1.20 RUBOUT

Octal data input routines will accept a RUBOUT command. Each time the RUBOUT key is pressed a — symbol is printed and a character is deleted. Typing two RUBOUTS will delete two characters etc. The rubout routine for octal values will back space only to the beginning of the field. Data is represented by one field (or byte) whereas addresses are represented by two bytes (fields). The routine will type a — for each RUBOUT until it reaches the beginning of the field where it will accept a RUBOUT but will not type any symbol and will not continue to back space.

SEC. 2 SOFTWARE LISTING

A software listing for the MONITOR 8 appears on the pages that follow.

```
                    RESET INDEX
RESET NØ.           FUNCTIØN
RST 000  CØLD START,GENERAL RESTART
RST 010  GØ TØ RØM 7 (FØR USER)
RST 020  ØUTPUT AN ASCII CHARACTER
RST 030  INPUT AN ASCII CHARACTER
RST 040  TEST FØR RUBØUT
RST 050  SEARCH FØR CHARACTER IN 'E' REGISTER
RST 060  BREAKPØINT EXECUTE
RST 070  TIMING LØØP


                    SUBRØUTINE INDEX
          (START ADDRESSES ØF MANY ØF THE RØUTINES USED HERE,
          WHICH MAY BE USABLE IN ØTHER SØFTWARE)

START ADDRESS       FUNCTIØN
000013    ØUTPUT CARRIAGE RETURN AND LINE FEED
000177    TEST FØR ØCTAL CHARACTER
000205    3 DIGIT ØCTAL INPUT (CØMPRESSED TØ 1 BYTE)
000253    3 DIGIT ØCTAL ØUTPUT (USED TØ DISPLAY 1 BYTE)
000311    ADDRESS INCREMENT (USES CLP-LØC 013377,013376)
000326    ADDRESS DECREMENT
000344    ADDRESS CØMPARE (CLP,CLP-1)
000362    CØMPARE AND INCREMENT (USED TØ TEST FØR END ØF RØUTINE)
001000    ØCTAL DUMP (DPØ)
001023    FETCH DATA FRØM LØCATIØN ADDRESSED BY CLP
001047    DISPLAY DATA AT CLP
001055    DISPLAY BLANK, CLP (ADDRESS)
001073    ØUTPUT CR/LF, CLP
001111    PUT DATA INTØ CLP
001120    ØCTAL INPUT (LDØ)
001200    INPUT AN ADDRESS (2 BYTES)
001236    ØCTAL EDITØR (EDT)
001336    INDIRECT JUMP
001353    CLEAR BREAKPØINT (CBP)
002000    PRØM PRØGRAMMING RØUTINE (PRG)
002110    SET UP CLP (LØC)
002115    DUMP IN BNPF FØRMAT (DBF)
002201    LØAD IN BNPF FØRMAT (LBF)
002257    BANK TØ BANK TRANSLATE (TRN)
002347    SET BREAKPØINT (SBP)
003000    CØNTRØLLER RØUTINE
003131    GENERAL ERRØR RØUTINE
003150    TABLE SEARCH
003244    BREAKPØINT EXECUTE
005063    REGISTER DECØDE
005313    PRINT 3 ASCII BYTES
000000/ 015 RST 010     INTERRUPT JAM INSTRUCTION
000001/ 006 XRA         CLEAR A
000002/ 125 OUT 013     IDLE PTR
000003/ 104 JMP 003000  GO TO CONTROLLER
000006/ 000 HLT         UNUSED LOCATIONS
000007/ 000 HLT
000010/ 104 JMP 000001  COLD START
```

modular micros martin research

```
000020/ 026  LCI 375     (RST 020)      Ø/P ØNE CHARACTER
000022/ 036  LDI 177     SET UP TIMING
000024/ 075  RST 070     IST BIT IS LØNGER
000025/ 104  JMP 000140 CØNTINUED ELSEWHERE
000030/ 006  LAI 001     (RST 030)      I/P CHARACTER
000032/ 127  ØUT 013     ENABLE PTR
000033/ 036  LDI 302     SET UP TIMING
000035/ 104 JMP 000371 GO WAIT FOR TTY CONTROLLER
000040/ 006  LAI 177     (RST 040)      RUBØUT TEST
000042/ 271  CPB
000043/ 013  RFZ         NØT RUBØUT SØ RETURN
000044/ 016  LBI 337     Ø/P ARRØW
000046/ 025  RST 020
000047/ 007  RET         FLAG SET TØ IGNØRE INPUT
000050/ 035  RST 030     (RST 050)      SEARCH FØR CHAR
000051/ 301  LAB         FETCH I/P      IN REG E
000052/ 274  CPE         CØMPARE
000053/ 053  RTZ         GØT CHAR
000054/ 104  JMP 000050 TRY NEXT ØNE
000057/ 377  HLT         UNUSED BYTE
000060/ 104  JMP 003244 (RST 060)      XQT BRKPT
000063/ 301  LAB                       I/P (CØNT)
000064/ 074  CPI 001     CNTRL A I/P
000066/ 013  RFZ         NØ- GØ AHEAD
000067/ 005  RST 000     YES- PANIC AND RESTART
000070/ 030  IND         (RST 070)      TIMING LØØP
000071/ 110  JFZ 000070 LØØPING
000074/ 007  RET         DØNE
000075/ 377  HLT         WAIT FØR I/P   I/P(CØNT)
000076/ 075  RST 070     TIME IST BIT
000077/ 250  XRA         CLEAR A REG
000100/ 127  ØUT 013     IDLE PTR FØR NØW
000101/ 125  ØUT 012     START Ø/P
000102/ 026  LCI 370     SET UP 8-BIT ASCII CODE
000104/ 036  LDI 171     SET UP 1-BIT DELAY
000106/ 075  RST 070     WAIT FØR IT
000107/ 101  INP 000     GET BIT
000110/ 054  XRI 377     CØMPLEMENT I/P
000112/ 125  ØUT 012     ECHØ TØ Ø/P
000113/ 032  RAR         RØTATE INTØ B
000114/ 301  LAB         WITH PREVIØUS
000115/ 032  RAR         BITS
000116/ 310  LBA
000117/ 020  INC         BUMP CØUNTER
000120/ 110  JFZ 000104 LØØP FØR MØRE BITS
000123/ 301  LAB         GØT 8 BITS NØW
000124/ 044  NDI 177     IGNØRE PARITY (MSB)
000126/ 310  LBA
000127/ 036  LDI 171     Ø/P STØP
000131/ 075  RST 070     AND Ø/P IDLE STATE
000132/ 006  LAI 001
000134/ 125  ØUT 012
000135/ 104  JMP 000063 TØ BE CØNTINUED
000140/ 020  INC                        Ø/P (CØNT)
000141/ 110  JFZ 000022 KEEP TIMING
000144/ 250  XRA         CLEAR A
000145/ 125  ØUT 012     START Ø/P
000146/ 026  LCI 370     SET UP TIMING
000150/ 036  LDI 171
000152/ 075  RST 070     WAIT FØR NEXT BIT
000153/ 301  LAB         FETCH BIT FRØM B
000154/ 125  ØUT 012     AND ØUTPUT BIT
000155/ 032  RAR         NØW SET UP THE NEXT
000156/ 310  LBA         BIT, STØRE IT IN B
000157/ 006  LAI 000
000161/ 032  RAR
000162/ 201  ADB
000163/ 310  LBA
000164/ 020  INC         BUMP CØUNT
000165/ 110  JFZ 000150 MØRE TØ Ø/P, SØ LØØP
000170/ 036  LDI 171     DØNE
000172/ 075  RST 070     Ø/P STØP AND IDLE BITS
000173/ 006  LAI 001
000175/ 125  ØUT 012
```

```
000176/ 007  RET          GØØDBYE
000177/ 035  RST 030      FETCH CHAR     TEST FØR ØCTAL
000200/ 044  NDI 370      MASK 3 BITS
000202/ 074  CPI 060      IS IT 06X? (ZF TELLS ALL)
000204/ 007  RET          GØ AWAY
000205/ 106  CAL 000177 GET DIGIT      ØCTAL I/P
000210/ 110  JFZ 000205 NØT ØCTAL,TRY AGAIN
000213/ 301  LAB          PUT DIGIT IN A
000214/ 012  RRC          RØTATE
000215/ 012  RRC          RØTATE
000216/ 370  LMA          STASH IT
000217/ 035  RST 030      FETCH DIGIT
000220/ 045  RST 040      TEST FØR RUBØUT
000221/ 150  JTZ 000205 TRY AGAIN
000224/ 307  LAM          FETCH LAST DIGIT
000225/ 044  NDI 300      MASK UNUSED BITS
000227/ 370  LMA          STØRE IT
000230/ 006  LAI 007      PUT IN 3 MØRE BITS
000232/ 241  NDB          RØTATE INTØ PØSITIØN
000233/ 002  RLC
000234/ 002  RLC
000235/ 002  RLC
000236/ 207  ADM          ADD IN THE ØLD DATA
000237/ 370  LMA          STØRE IT
000240/ 035  RST 030      FETCH DIGIT    ENTRY FØR
000241/ 045  RST 040      TEST           CØRRECTIØN DØNE
000242/ 150  JTZ 000217 RUBØUT
000245/ 006  LAI 007      MASK ALL BUT 3 BITS
000247/ 241  NDB
000250/ 207  ADM          ADD THE PREVIØUS BITS
000251/ 370  LMA          STASH DATA
000252/ 007  RET          DØNE
000253/ 016  LBI 240      (BLANK)        ØCTAL Ø/P
000255/ 025  RST 020                     3 DIGITS
000256/ 307  LAM          FETCH BYTE
000257/ 002  RLC          MØVE BITS 7 AND 8
000260/ 002  RLC          TØ PØS 1 AND 2
000261/ 044  NDI 003      MASK THE REST
000263/ 004  ADI 260      CØNVERT TØ ASCII
000265/ 310  LBA          SET UP FØR Ø/P
000266/ 025  RST 020      Ø/P
000267/ 307  LAM          FECTH BYTE
000270/ 012  RRC          SET UP BITS 4,5,6
000271/ 012  RRC
000272/ 012  RRC
000273/ 044  NDI 007      MASK
000275/ 004  ADI 260      CØNVERT
000277/ 310  LBA
000300/ 025  RST 020      Ø/P
000301/ 307  LAM          BITS 1,2,3 THIS TIME
000302/ 044  NDI 007
000304/ 004  ADI 260
000306/ 310  LBA          NØW Ø/P
000307/ 025  RST 020
000310/ 007  RET          ALL DØNE
000311/ 066  LLI 376      SET TØ CLP     ADDRESS INCR
000313/ 056  LHI 013                     2 BYTES
000315/ 317  LBM          FETCH
000316/ 010  INB          INCR LSB
000317/ 371  LMB          STØRE
000320/ 013  RFZ          CARRY
000321/ 060  INL          YES-INCR MSB
000322/ 317  LBM          FETCH
000323/ 010  INB          INCR
000324/ 371  LMB          STØRE
000325/ 007  RET          DØNE
000326/ 066  LLI 376      SET TØ CLP     ADDRESS DECR
000330/ 056  LHI 013                     2 BYTES
000332/ 317  LBM          FETCH
000333/ 011  DCB          DECR
000334/ 371  LMB          STØRE
000335/ 060  INL          PØINT TØ MSB
000336/ 010  INB          WAS LSB
000337/ 013  RFZ          NØ-RETURN
```

```
000340/ 317  LBM          YES-D0 MSB
000341/ 011  DCB
000342/ 371  LMB
000343/ 007  RET          D0NE
000344/ 066  LLI 377      SET T0 CLP     ADDRESS C0MP
000346/ 056  LHI 013                     4 BYTES
000350/ 307  LAM          FETCH MSB
000351/ 061  DCL
000352/ 317  LBM          FETCH LSB
000353/ 061  DCL
000354/ 277  CPM          C0MPARE MSB
000355/ 013  RFZ          N0T EQUAL-RETURN
000356/ 301  LAB          PUT LSB INT0 A
000357/ 061  DCL
000360/ 277  CPM          C0MPARE
000361/ 007  RET          G0 AWAY (ZF=1 IF EQUAL)
000362/ 106  CAL 000344 C0MP ADDR      C0MP ADDR AND
000365/ 110  JFZ 000311 INCR ADDR      INCR IF N0T =
000370/ 005  RST 000     N0W RESTART
000371/ 101  INP 000     INPUT TTY BIT
000372/ 032  RAR         SHIFT INTO CARRY POSITION
000373/ 100  JFC 000371 TEST F0R CHARACTER
000376/ 104  JMP 300076 CONTINUE IF START BIT FOUND
001000/ 300  LAA         NOP


                                         0CTAL DUMP (DP0)
001001/ 046  LEI 010     SET UP CHAR/LINE
001003/ 106  CAL 001073 0/P CLP
001006/ 106  CAL 001047 0/P CLP C0NTENTS
001011/ 106  CAL 000362 INCR AND C0MPARE CLP
001014/ 041  DCE         INCR LINE C0UNT
001015/ 150  JTZ 001001 NEW LINE,PRINT ADR
001020/ 104  JMP 001006 SAME LINE, JUST L00P
001023/ 066  LLI 377     SET T0 CLP     GET DATA (FR0M CLP)
001025/ 056  LHI 013                    EXTENDED ADDRESS
001027/ 307  LAM         MS IN 'A'      PUTS DATA INT0 013370
001030/ 061  DCL
001031/ 367  LLM         LS IN L
001032/ 350  LHA         N0W PUT MS IN H
001033/ 002  RLC         R0TATE T0 TEST BIT 8
001034/ 307  LAM         FETCH LS
001035/ 003  RFC         RETURN IF N0T EXTENDED MEM0RY
001036/ 056  LHI 013     GET DATA FR0M I/P P0RT
001040/ 306  LAL         SET TEMP ST0RE L0CATI0N
001041/ 066  LLI 370
001043/ 121  0UT 010     0/P LS
001044/ 103  INP 001     GET DATA
001045/ 370  LMA         PUT INT0 MEM0RY
001046/ 007  RET         G01
001047/ 106  CAL 001023                  GET DATA FR0M CLP
001052/ 104  JMP 000253                  AND PRINT IT
001055/ 066  LLI 377     SET T0 CLP     0/ ' HHHLLL'
001057/ 056  LHI 013
001061/ 016  LBI 240     0/P BLANK
001063/ 025  RST 020
001064/ 106  CAL 000256 0/P MS BYTE
001067/ 061  DCL         0/P LS BYTE
001070/ 104  JMP 000256 AND RETURN T0 CALL WHEN D0NE
001073/ 066  LLI 377     SET T0 CLP     PRINT CR/LF HHHLLL
001075/ 056  LHI 013
001077/ 106  CAL 000013 0/P CR/LF
001102/ 106  CAL 001064 0/P ADR
001105/ 016  LBI 257     0/P SLASH
001107/ 025  RST 020
001110/ 007  RET         D0NE
001111/ 106  CAL 001023 SET T0 CLP     GET CLP PUT DATA THERE
001114/ 104  JMP 000205 FETCH DATA
001117/ 300  LAA         N0P N0T USED
001120/ 106  CAL 000013 0/P CR/LF      0CTAL INPUT (LD0)
001123/ 046  LEI 057     SEARCH F0R SLASH (/)
001125/ 055  RST 050
001126/ 035  RST 030     FETCH CHARACTER
001127/ 301  LAB
```

modular micros
martin research

```
001130/ 074  CPI 015     IS IT A CR
001132/ 150  JTZ 001123  YES- WAIT FØR AN0THER SLASH
001135/ 106  CAL 001111  N0- PUT DATA AT CLP
001140/ 106  CAL 000362  C0MPARE AND INCR CLP
001143/ 104  JMP 001126  L00P
001146/ 066  LLI 373     SET UP 'L'    C0PY (CPY)
001150/ 106  CAL 001205  INPUT NEW START 0F BL0CK
001153/ 106  CAL 001023  SET UP H AND L
001156/ 327  LCM         FETCH DATA
001157/ 066  LLI 373
001161/ 106  CAL 001025  SET H,L T0 NEW ADR
001164/ 372  LMC         ST0RE DATA
001165/ 106  CAL 000362  INCR FR0M ADR
001170/ 066  LLI 372
001172/ 106  CAL 000313  INCR T0 ADR
001175/ 104  JMP 001153  L00P
001200/ 066  LLI 377     SET CLP        GET ADDRESS (2 BYTES)
001202/ 106  CAL 000013  0/P CR/LF
001205/ 016  LBI 252     0/P *
001207/ 025  RST 020
001210/ 056  LHI 013     CLP PAGE
001212/ 106  CAL 000205  GET A BYTE (MS)
001215/ 061  DCL         SET  L FØR LS
001216/ 035  RST 030     GET NEXT BYTE
001217/ 045  RST 040     RUB-0UT?
001220/ 110  JFZ 000213  N0-GET THE NEW BYTE AS BEFØRE
001223/ 060  INL         YES-RESTØRE L T0 MS ADR
001224/ 307  LAM         FETCH MS BYTE
001225/ 044  NDI 370     MASK 3 BITS
001227/ 370  LMA         ST0RE
001230/ 106  CAL 000240  GET 3 NEW BITS
001233/ 104  JMP 001215  N0W- THE LS BYTE
001236/ 106  CAL 001073  CR/LF+CLP      0CTAL EDIT0R (EDT)
001241/ 106  CAL 001247  PR0CESS LINE (BYTE)
001244/ 104  JMP 001236  L00P
001247/ 035  RST 030     FETCH I/P
001250/ 301  LAB
001251/ 074  CPI 122     TEST FØR 'R'
001253/ 150  JTZ 003014  YES-THEN RETURN
001256/ 074  CPI 052     TEST FØR '*'
001260/ 066  LLI 377     SET L T0 CLP
001262/ 150  JTZ 002110  G0 T0 L0C R0UTINE
001265/ 074  CPI 100     TEST FØR  @
001267/ 150  JTZ 003320  G0 T0 XQT
001272/ 074  CPI 136     TEST FØR 'UP ARR0W'
001274/ 150  JTZ 000326  THEN DECR CLP
001277/ 074  CPI 040     TEST FØR BLANK
001301/ 150  JTZ 001321  PRINT THIS BYTE
001304/ 106  CAL 000200  FAILED ALL TESTS, IS I/P 0CTAL?
001307/ 013  RFZ         N0- IGN0RE IT
001310/ 106  CAL 001023  YES-SET H AND L
001313/ 106  CAL 000213  GET 2 M0RE DIGITS AND ST0RE THE BYTE
001316/ 104  JMP 000311  INCR CLP AND L00P
001321/ 106  CAL 001047  FETCH AND PRINT DATA
001324/ 035  RST 030     I/P M0RE
001325/ 301  LAB         T0 'A' REG
001326/ 074  CPI 137     IS IT BACK ARR0W
001330/ 152  CTZ 000205  YES-REPLACE DATA BYTE
001333/ 104  JMP 000311  INCR CLP AND L00P
001336/ 066  LLI 371                  INDIRECT JUMP
001340/ 056  LHI 013     SET H,L T0 UNUSED RAM
001342/ 076  LMI 104     ST0RE 'JMP'
001344/ 060  INL
001345/ 371  LMB         LS ADR IN 'B'
001346/ 060  INL
001347/ 370  LMA         MS ADR IN 'A'
001350/ 104  JMP 013371  G0 JMP IN
001353/ 066  LLI 365                  CLEAR BREAKP0INT (CBP)
001355/ 056  LHI 013     3 BYTES FØR BRKPT P0INTERS
001357/ 347  LEM         WHAT WAS INSTR
001360/ 060  INL
001361/ 060  INL
001362/ 106  CAL 001027  SET H AND L
001365/ 074  CPI 065     NO BREAKPOINT SET
```

```
001367/ 013  RFZ          YES--RETURN UNTOUCHED
001370/ 374  LME          NO--CLEAR BREAKPOINT
001371/ 007  RET
001372/ 000  HLT          UNUSED LOCATIONS
001373/ 000  HLT
001374/ 000  HLT
001375/ 000  HLT
001376/ 000  HLT
001377/ 000  HLT
002000/ 016  LBI 045      TYPE  Z      PRØGRAMMER (PRG)
002002/ 025  RST 020
002003/ 035  RST 030      INPUT TIMING CØNSTANT
002004/ 002  RLC          GIVING A= 005 (1602A/1702A)
002005/ 002  RLC               N= 071 (1602/1702)
002006/ 340  LEA          SAVE AWAY FØR LATER
002007/ 106  CAL 001023   GET DATA AND SET H AND L
002012/ 306  LAL          GET PRØM ADDRESS
002013/ 121  ØUT 010      AND ØUTPUT TØ PRØGRAMMER
002014/ 103  INP 001      GET RØM DATA
002015/ 277  CPM          AND IF NØT EQUAL
002016/ 112  CFZ 002027   GØ PRØGRAM
002021/ 106  CAL 000362   INCREMENT ADDRESS PØINTER
002024/ 104  JMP 002007   AND GØ BACK TØ TEST NEXT BYTE
002027/ 016  LBI 001      START WITH 1 TRY   PRØG. SEQUENCE
002031/ 106  CAL 002051   GØ PRØGRAM IT
002034/ 301  LAB          DATA IS NØW READ AS CØRRECT
002035/ 002  RLC          SØ ØVERKILL 4 TIMES
002036/ 002  RLC
002037/ 310  LBA          B STILL CØUNTS TRIES
002040/ 106  CAL 002051   GØ ØVERKILL
002043/ 011  DCB          UNTIL B
002044/ 110  JFZ 002040   EQUALS ZERØ
002047/ 025  RST 020      AND ØUTPUT  NULL CHARACTER
002050/ 007  RET          INDICATING END ØF BYTE
002051/ 307  LAM          GET DATA
002052/ 054  XRI 377      CØMPLEMENT IT
002054/ 123  ØUT 011      AND PUT IT IN THE BUFFER
002055/ 006  LAI 004      SET UP FØR THE 15MSEC
002057/ 127  ØUT 013      PULSE GENERATØR
002060/ 250  XRA          HIT IT ØNCE
002061/ 127  ØUT 013      GIVING ØNE  3.0 MSEC PULSE
002062/ 324  LCE          E STILL HAS TIMING CØNSTANT
002063/ 036  LDI 325      INNER TIME-ØUT
002065/ 075  RST 070      LØØP
002066/ 021  DCC          TØTAL LØØP TIME IS
002067/ 110  JFZ 002063   15 MSEC ØR 160 MSEC
002072/ 103  INP 001      HØWS THE DATA LØØK?
002073/ 277  CPM
002074/ 053  RTZ          IF THE SAME RETURN
002075/ 010  INB          B CØUNTS UNSUCCESSFUL TRIES
002076/ 110  JFZ 002051   IF NØT 377 TRIES TRY AGAIN
002101/ 106  CAL 001055   PRINT CURRENT LØCATIØN PØINTER
002104/ 016  LBI 277      AND THEN A  ?
002106/ 025  RST 020      AND GIVE UP BY DØING
002107/ 005  RST 000      A CØMPLETE RSTART
002110/ 066  LLI 377      ADR ØF CLP     SET CLP  (LØC)
002112/ 104  JMP 003143   I/P ADR, RET HØME
002115/ 300  LAA          NOP     BNPF DUMP--DBF
002116/ 066  LLI 371      SCRATCH LØCATIØN
002120/ 076  LMI 005      Ø/P 5 BYTES PER LINE
002122/ 016  LBI 240      NØW, Ø/P A BLANK
002124/ 025  RST 020
002125/ 106  CAL 001023   GET DATA
002130/ 360  LLA          SAVE IT IN L
002131/ 016  LBI 302      Ø/P 'B'
002133/ 025  RST 020
002134/ 046  LEI 010      8 BITS PER BYTE
002136/ 306  LAL          RØTATE DATA IN 'L'
002137/ 002  RLC          PUT NEXT BIT IN CARRY
002140/ 360  LLA
002141/ 016  LBI 316      SET 'B' TØ 'N'
002143/ 100  JFC 002150   IF BIT IS 0, JUMP
002146/ 016  LBI 320      BIT =1 SØ CHANGE TØ 'P'
002150/ 025  RST 020      Ø/P WHATEVER IT IS
002151/ 041  DCE          ØNE MØRE BIT DØNE
002152/ 110  JFZ 002136   LØØP IF MØRE
```

```
002155/ 016  LBI 306     DØNE BYTE,Ø/P 'F'
002157/ 025  RST 020
002160/ 106  CAL 000362 INCR,CØMP CLP
002163/ 066  LLI 371     SET UP 'L' AGAIN
002165/ 317  LBM         ØNE MØRE BYTE Ø/P
002166/ 011  DCB
002167/ 371  LMB
002170/ 110  JFZ 002122 MØRE ØN THIS LINE
002173/ 106  CAL 000013 NEW LINE (CR/LF)
002176/ 104  JMP 002116 KEEP GØING
002201/ 300  LAA         NØP             BNPF LØAD (LBF)
002202/ 046  LEI 102     WAIT FØR A 'B'
002204/ 055  RST 050
002205/ 046  LEI 370     NØW 8 BITS EXPECTED
002207/ 106  CAL 001023 SET H,L
002212/ 076  LMI 000     CLEAR SØME RAM
002214/ 035  RST 030     FETCH CHARACTER
002215/ 301  LAB         INTØ 'A'
002216/ 074  CPI 116     IS IT 'N'
002220/ 150  JTZ 002232 YES- STASH IT
002223/ 054  XRI 377     NØ -CØMPLEMENT
002225/ 074  CPI 257     IS IT 'P'
002227/ 110  JFZ 002101 NØ-ERRØR
002232/ 032  RAR         YES- PUT BIT IN CARRY
002233/ 307  LAM         GET PREVIØUS BITS
002234/ 022  RAL         RØTATE IN NEW BIT
002235/ 370  LMA         STASH IT
002236/ 040  INE         CØUNT YØUR BITS
002237/ 110  JFZ 002214 NØT DØNE,LØØP
002242/ 035  RST 030     YES-ØNE MØRE CHECK
002243/ 301  LAB
002244/ 074  CPI 106     LAST CHARACTER MUST BE AN 'F'
002246/ 110  JFZ 002101 NØ-PANIC
002251/ 106  CAL 000362 YES-INCR CLP,CHECK IF DØNE
002254/ 104  JMP 002202 LØØP IF YØU GET HERE
002257/ 066  LLI 373                  BANK TØ BANK TRANSLATE(TRN)
002261/ 106  CAL 000205 FETCH ØLD BANK NØ.
002264/ 016  LBI 337     Ø/P BACK ARRØW
002266/ 025  RST 020
002267/ 061  DCL         FETCH NEW BANK NØ.
002270/ 106  CAL 000205
002273/ 106  CAL 001023 GET DATA (INSTR)
002276/ 347  LEM
002277/ 106  CAL 006320 IS IT 1,2 ØR 3 BYTE INSTR
002302/ 340  LEA         'A' HAS PØINTER (Ø=1BYTE)
002303/ 106  CAL 000362 INCR CLP        (1=2BYTE)
002306/ 304  LAE         RØTATE PØINTER  (3=3BYTE)
002307/ 012  RRC
002310/ 140  JTC 002302 LØØP FØR MØRE
002313/ 074  CPI 140     WAS IT A 3 BYTE INSTR (JMP ØR CAL)
002315/ 110  JFZ 002273 NØ-GØ TØ NEXT BYTE
002320/ 106  CAL 001023 YES-SET UP H,L
002323/ 061  DCL         TØ LAST BYTE ØF JMP ØR CAL
002324/ 307  LAM         FETCH BYTE
002325/ 056  LHI 013     WAS IT ØUR MAGIC NØ.?
002327/ 066  LLI 373
002331/ 277  CPM
002332/ 110  JFZ 002273 NØ-GØ AWAY
002335/ 061  DCL         YES-GET THE NEW ØNE
002336/ 347  LEM
002337/ 106  CAL 001023 SET UP H,L
002342/ 061  DCL         (LS-1) ØF CØURSE
002343/ 374  LME         REPLACE MS BYTE
002344/ 104  JMP 002273 NØW- WE ARE REALLY DØNE
002347/ 106  CAL 001353 CLEAR ØLD     SET BREAKPØINT (SBP)
002352/ 106  CAL 001200 FETCH ADR ØF NEW BRKPT
002355/ 106  CAL 001023 SET UP H,L TØ CLP
002360/ 326  LCL         SAVE H,L
002361/ 335  LDH
002362/ 076  LMI 065     SET RST 060 INTØ LØCATIØN
002364/ 056  LHI 013     SAVE THE ØLD INSTR
002366/ 066  LLI 365
002370/ 370  LMA         IT WAS LEFT IN 'A'
002371/ 060  INL
```

modular micros martin research

```
002372/ 372  LMC          'L' (LS ADR)
002373/ 060  INL
002374/ 373  LMD          'H' (MS ADR)
002375/ 104  JMP 003000 GØ HØME TØ MØMMY
003000/ 106  CAL 000013 CR/LF          MØNITØR AND CØNTRØLLER
003003/ 046  LEI 010    SET UP LØØP
003005/ 016  LBI 255    CHARACTER IS '-'
003007/ 025  RST 020    GØ PRINT '--------'
003010/ 041  DCE        CØUNT
003011/ 110  JFZ 003007 LØØP
003014/ 146  CAL 000013 NEW LINE
003017/ 146  CAL 003067 FETCH INPUT
003022/ 100  JFC 003017 LØØP IF NØT 'A'-'Z'
003025/ 146  CAL 003100 NØW GET TWØ MØRE CHARACTERS
003030/ 146  CAL 003150 FIND IT IN THE TABLE
003033/ 150  JTZ 006000 NØT FØUND!! GØ TØ LDS
003036/ 066  LLI 373                   EXEC RØUTINE
003040/ 056  LHI 013
003042/ 370  LMA        STØRE ADDRESS (MS)
003043/ 061  DCL        FRØM 5 BYTE TABLE
003044/ 371  LMB        STØRE ADDRESS (LS)
003045/ 044  NDI 200    FRØM 5 BYTE TABLE
003047/ 112  CFZ 003137 GØ FETCH INITIAL AND FINAL ADDRESS
003052/ 112  CFZ 000013 IF MS=1XXXXXXX, START WITH CR/LF
003055/ 066  LLI 371    JMP IN 371
003057/ 076  LMI 104    YES IT IS AN INDIRECT JMP
003061/ 106  CAL 013371 SØ GØ
003064/ 144  JMP 003014 AND CØNTINUE WHEN DØNE
003067/ 035  RST 030    GET CHAR       CHAR TEST
003070/ 006  LAI 100    TEST FØR  LT 'A'
003072/ 271  CPB
003073/ 301  LAB
003074/ 003  RFC        PASS IF GT ØR EQ 'A'
003075/ 074  CPI 133    TEST IF GT 'Z'
003077/ 047  RET        GØ, CARRY TELLS ALL
003100/ 046  LEI 002                   SYM INPUT
003102/ 066  LLI 350    CHAR IN 013350
003104/ 056  LHI 013
003106/ 370  LMA        STØRE
003107/ 146  CAL 003067 TEST NEXT CHAR
003112/ 100  JFC 003131 NØT 'A'-'Z' !! ERRØR!
003115/ 060  INL        SET UP NEXT ØNE
003116/ 041  DCE        CØUNT
003117/ 110  JFZ 003106 NØT DØNE, LØØP
003122/ 370  LMA        DØNE- STØRE LAST CHAR
003123/ 340  LEA        NØW SET UP REG, 3 GØES IN 'E'
003124/ 061  DCL
003125/ 337  LDM        2 IN 'D'
003126/ 061  DCL
003127/ 327  LCM        AND 1 IN 'C'
003130/ 007  RET        ALL DØNE
003131/ 016  LBI 277                   ERRØRS CØME HERE
003133/ 025  RST 020    PRINT '?'
003134/ 144  JMP 003014 GØ GET ANØTHER INPUT
003137/ 106  CAL 001200 CR/LF AND '*' INITIAL AND FINAL ADR
003142/ 061  DCL        GØT FIRST ADR
003143/ 016  LBI 240    PRINT BLANK
003145/ 104  JMP 001207 GET FINAL ADR,GØ BACK HØME
003150/ 016  LBI 022                   SEARCH TABLE
003152/ 066  LLI 021    5 BYTE TABLE
003154/ 056  LHI 004
003156/ 307  LAM        NØW GET 1ST CHAR
003157/ 060  INL        READY FØR NEXT CHAR
003160/ 272  CPC        CØMPARE TABLE AND I/P
003161/ 110  JFZ 003204 JMP IF NØT EQUAL
003164/ 307  LAM        GET 2ND
003165/ 060  INL        READY FØR 3RD
003166/ 273  CPD        CØMPARE
003167/ 110  JFZ 003205 JMP IF NØT THE SAME
003172/ 307  LAM        NØW FØR THE 3RD
003173/ 060  INL        AND PREPARE FØR DATA
003174/ 274  CPE        CØMPARE AS BEFØRE
003175/ 110  JFZ 003206 AND JUMP IF NØT NICE
003200/ 317  LBM        GET 'GØ TØ' ADDRESS
```

```
003201/ 060  INL
003202/ 307  LAM         2 BYTES ØF IT
003203/ 007  RET         AND RETURN
003204/ 060  INL                      LØØK AT NEXT SYMBØL
003205/ 060  INL                      IN THE TABLE
003206/ 060  INL
003207/ 060  INL
003210/ 011  DCB         CØUNT ØUR TRYS
003211/ 053  RTZ         END ØF TABLE
003212/ 104  JMP 003156 MØRE TØ CHECK
003215/ 056  LHI 004                  3BYTE TABLE SEARCH
003217/ 021  DCC         'C' IS CØUNTER
003220/ 053  RTZ         RETURN WHEN DØNE TABLE
003221/ 307  LAM         NØW LØØK AT THE FIRST ENTRY
003222/ 060  INL         CØMPARE WITH DATA
003223/ 273  CPD         JMP IF NØT LIKED
003224/ 110  JFZ 003237 2ND ENTRY AS ABØVE
003227/ 307  LAM
003230/ 274  CPE
003231/ 110  JFZ 003237 AND JUMP, MAYBE
003234/ 060  INL         FETCH DATA FRØM TABLE
003235/ 307  LAM
003236/ 007  RET         RETURN TØ LDS RØUTINE
003237/ 060  INL         NEXT ENTRY
003240/ 060  INL         EØØP AND TRY AGAIN
003241/ 104  JMP 003217 LØØP AND TRY AGAIN
003244/ 345  LEH                       BRKPT EXECUTE
003245/ 336  LDL         SAVE L,H  LØSING D,E
003246/ 066  LLI 364     SAVE REGISTERS A-E
003250/ 056  LHI 013     IN RAM (LØC 013364 TØ 013360)
003252/ 374  LME
003253/ 061  DCL
003254/ 373  LMD
003255/ 061  DCL
003256/ 372  LMC
003257/ 061  DCL
003260/ 371  LMB
003261/ 061  DCL
003262/ 370  LMA
003263/ 006  LAI 030     NØW DISPLAY CARRY FLAG
003265/ 022  RAL         RØTATE IN CARRY AND CØNVERT TØ
003266/ 340  LEA         ASCII
003267/ 016  LBI 240     Ø/P BLANK
003271/ 025  RST 020
003272/ 314  LBE         Ø/P CARRY FLAG
003273/ 025  RST 020
003274/ 046  LEI 005     SET UP CØUNT TØ PRINT REGISTERS
003276/ 066  LLI 360     START ØF SAVED REGISTERS
003300/ 106  CAL 000253 PRINT BYTE AS ØCTAL
003303/ 060  INL         NEXT REGISTER
003304/ 041  DCE         CØUNT
003305/ 110  JFZ 003300 LØØP TILL DØNE
003310/ 061  DCL         GØ BACK ØNE REG
003311/ 106  CAL 001027 AND GET DATA AT H,L LØCATIØN
003314/ 106  CAL 000253 AND PRINT IT
003317/ 005  RST 000     NØW WERE DØNE, GØ HØME
003320/ 066  LLI 373                   XQT RØUTINE
003322/ 106  CAL 003143 LØAD ADDRESS
003325/ 104  JMP 003052 EXEC WILL SEND US THERE

                         RØM NUMBER 3 CØNTINUED WITH DPS RØUTINES
                         RØM 4 CØNTAINS THE SYMBØL TABLES,
                         AS FØLLØWS:
```

1. 5 BYTE TABLE

THE 5 BYTE TABLE ØCCUPIES PØSITIØNS 004021 TØ 004157 INCLUSIVE AND CØNTAINS ALL MØNITØR CØMMANDS PLUS THE MACHINE CØMMANDS HLT, INP,ØUT,RST, AND THE SPECIAL SYMBØL ???, INDICATIING A NØ FIND CØNDITIØN ØN ØUTPUT. THE INPUT RØUTINE DØES NØT USE THIS SYMBØL. THE FØRMAT IS THUS:

ASCII	X	Q	T	(DATA FIELD)
ØCTAL	130	121	124	320 003

modular micros
martin research

```
ADDRESS 021  022  023  024   025
```

WHEN A FIND IS MADE DURING A SEARCH, THE DATA FIELD IS MØVED TØ REGISTERS A AND B, AND AN INDIRECT JUMP MADE TØ THAT ADDRESS, IF THE MS HALF ØF THE ADDRESS IF A 2XX, THE EXEC WILL LØØK FØR TWØ ADDRESSES BEFØRE GØING TØ THE RØUTINE.

DURING A SYMBØLIC DUMP, THE LAST 5 SYMBØLS ARE USED FØR THE APPRØPRIATE MACHINE CØMMANDS, AND ARE STØRED AS ØUTPUT.

```
ADR     SYMB LS  MS (ADDRESS ØF RØUTINE)
004021/  XQT 320 003
004026/  EDT 236 001
004033/  LDØ 120 201
004040/  LBF 201 202
004045/  DPO 001 201
004052/  DBF 116 202
004057/  DPS 000 205
004064/  CPY 146 201
004071/  TRN 257 202
004076/  SBP 347 002
004103/  CBP 353 001
004110/  PRG 000 202
004115/  LØC 110 002
004122/  DLP 055 001
004127/  HLT 046 006
004134/  RST 270 006
004141/  INP 270 006
004146/  ØUT 270 006
004153/  ???
```

2. 3 BYTE TABLE

THIS TABLE CØNTAINS TWØ BYTES ØF ASCII CØDE AND ØNE DATA BYTE, WHICH IS A MASKED PØRTIØN ØF THE INSTRUCTIØN. THE FØRMAT IS:

```
ASCII     N          D          (DATA)
ØCTAL     116        104        244
LØCATIØN  252        253        254
```

THE TABLE ØCCUPIES LØCATIØNS 004156 TØ 004273,AND IS USED IN TWØ WAYS. THE LDS RØUTINE CØMPARES THE TWØ ASCII CHARACTERS TØ THE INPUT CHARACTERS, AND RETURNS THE DATA IN THE A REGISTER IF A FIND IS MADE. FØR THE DPS RØUTINE, THE PARTIAL WØRD (DATA) IS TESTED, AND THE H AND L REGISTERS ARE USED TØ RETRIEVE THE ASCII AS NEEDED.

3 BYTE TABLE:

```
LØCATIØN ASCII     DATA
004156   LC        002
004161   RC        012
004164   AL        022
004167   AR        032
004172   JMP       104       (JMP)
004176   CAL       106       (CAL)
004202   RET       007       (RET)
004206    TC       040
004211    FC       000
004214    TZ       050
004217    FZ       010
004222    TS       060
004225    FS       020
004230    TP       070
004233    FP       030
004236    AD       204
004241    AC       214
004244    SU       224
004247    SB       234
004252    ND       244
004255    XR       254
004260    ØR       264
004263    CP       274
004266    IN       000
004271    DC       001
```

3. 4 BYTE TABLE

THE 4 BYTE TABLE ØCCUPIES PØSITIØNS 004274 TØ 004377, AND IS USED BY THE DPS RØUTINE.
THE FØRMAT IS:

	MASK	DATA	ADDRESS	DATA FIELD
	361	101	161	144
LØC	310	311	312	313

THE MASK CHARACTER IS USED TØ MASK (AND) DØN'T CARE BITS IN THE INPUT BYTE, THE REMAINING BITS ARE CØMPARED TØ THE DATA IN THE NEXT FIELD TØ DECØDE AN INSTRUCTIØN. IF A FIND IS MADE THE ADDRESS IS USED FØR AN INDIRECT JUMP (TØ 005AAA). THE LAST ENTRY IS AN UNCØNDITIØNAL FIND WHICH ØUTPUTS THE ERRØR SYMBØL ???.
THE DATA FIELD CØLUMN IS USED FØR VARIØUS PURPØSES BY THE CALLED RØUTINES.

4 BYTE TABLE

LØCATIØN	MASK	DATA	ADDRESS	DATA FIELD
004274	377	377	155	132
004300	376	000	155	132
004304	376	070	155	156
004310	361	101	161	144
004314	347	002	251	037
004320	307	006	262	352
004324	307	005	161	137
004330	307	004	125	111
004334	307	001	142	273
004340	307	000	142	270
004344	303	003	215	202
004350	303	102	215	176
004354	301	101	161	151
004360	303	100	215	172
004364	300	300	272	000
004370	300	200	120	000
004374	000	000	155	156

SYMBØLIC RØUTINES

NØTE: THESE RØUTINES CØVER PART ØF RØM 3,4 AND ALL ØF RØMS 5,6

```
003330/ 106  CAL 005352 GET 3 BYTES   DPS ØUTPUT
003333/ 106  CAL 005104 LØAD THEM INTØ REGISTERS
003336/ 106  CAL 005313 ØUTPUT THEM
003341/ 347  LEM        LØAD E WITH DATA
003342/ 106  CAL 006320 DECØDE LENGTH
003345/ 012  RRC        1 BYTE INSTR?
003346/ 100  JFC 005363 YES-GØ TØ LINE CHECK
003351/ 340  LEA        SAVE LENGTH BITS
003352/ 106  CAL 000362 INCR ADR
003355/ 106  CAL 001023 GET DATA
003360/ 041  DCE        3 BYTES MAYBE?
003361/ 160  JTS 003372 SIGN FLAG =1 IF SØ
003364/ 106  CAL 000253 Ø/P IMMEDIATE DATA
003367/ 104  JMP 005363 AND GØ TØ LINE CHECK
003372/ 327  LCM        YES ITS 3 BYTE! GET LS ADR
003373/ 106  CAL 000362 INC CLP
003376/ 312  LBC        MØVE ADR TØ B
003377/ 300  LAA        NØP (UNUSED BYTE)
004000/ 106  CAL 001023 GET DATA (MS ADR BYTE)
004003/ 327  LCM        SAVE IN C
004004/ 106  CAL 005104 LØAD 3 BYTES
004007/ 060  INL        SET UP DATA PØINTERS
004010/ 060  INL
004011/ 106  CAL 001061 ØUTPUT THIS ADRESS
004014/ 104  JMP 005363 AND GØ ØN TØ LINE CHECK
005000/ 016  LBI 012                SYMBØLIC DUMP (DPS)
005002/ 025  RST 020    PRINT 3 LF'S
005003/ 025  RST 020
005004/ 025  RST 020
005005/ 046  LEI 076    SET UP LINES/PAGE
005007/ 066  LLI 353    AND STØRE NUMBER AT 013353
```

modular micros
martin research

```
005011/ 056  LHI 013
005013/ 374  LME
005014/ 106  CAL 001073 GET CLP AND PRINT IT
005017/ 106  CAL 001047 GET DATA AND PRINT IT
005022/ 347  LEM        SAVE DATA IN 'E'
005023/ 337  LDM        AND IN 'D'
005024/ 106  CAL 005063 ASSUME BITS 3-5 ARE A DESTINATIØN REG.
005027/ 061  DCL        STØRE IT IN 013351,013352
005030/ 370  LMA
005031/ 066  LLI 274    SET UP START ØF 4 BYTE TABLE
005033/ 056  LHI 004
005035/ 303  LAD        GET MASK FRØM TABLE
005036/ 247  NDM        AND MASK DØNT CARE BITS
005037/ 060  INL        NØW CHECK THE REST
005040/ 277  CPM        WITH THE TABLE
005041/ 110  JFZ 005055 JUMP IF NØ FIND
005044/ 006  LAI 005    LØAD MS BYTE ØF ADR
005046/ 060  INL
005047/ 317  LBM        LØAD LS BYTE ØF ADR
005050/ 060  INL
005051/ 327  LCM        LØAD C WITH DATA FRØM TABLE
005052/ 104  JMP 001336 AND DØ AN INDIRECT JUMP TØ RØUTINE
005055/ 060  INL        (NØ FIND) INCR L TØ
005056/ 060  INL        NEXT TABLE ENTRY
005057/ 060  INL
005060/ 104  JMP 005035 GØ LØØP
005063/ 303  LAD        GET DATA      REGISTER DECØDE
005064/ 012  RRC
005065/ 012  RRC        LØØK AT BITS 3-5
005066/ 012  RRC
005067/ 044  NDI 007    MASK THE REST
005071/ 004  ADI 370    AND ADD START ØF TABLE
005073/ 360  LLA        TABLE ADR TØ 'L'DR
005074/ 056  LHI 006    MS ADR ØF TABLE
005076/ 307  LAM        GET REGISTER
005077/ 066  LLI 352
005101/ 104  JMP 005114 DØNE
005104/ 066  LLI 352    SET DP        3 BYTE LØAD
005106/ 056  LHI 013
005110/ 372  LMC        SAVE C
005111/ 061  DCL
005112/ 371  LMB        SAVE B
005113/ 061  DCL        L=350 NØW
005114/ 056  LHI 013    ENTRY FØR 1 BYTE LØAD
005116/ 370  LMA        AND SAVE A
005117/ 007  RET        GØ AWAY SØMWHERE
005120/ 303  LAD                      ACC GRØUP RØUTINE
005121/ 106  CAL 005067 DECØDE SØURCE REG
005124/ 320  LCA        AND PUT IN C
005125/ 303  LAD                      ENTRY FØR IMMEDIATE
005126/ 044  NDI 070    MASK ØUT SØURCE PART
005130/ 004  ADI 204    (SPECIALLY FØR 'I' INSTR)
005132/ 066  LLI 240    START ØF ACC IN 3 BYTE TABLE
005134/ 106  CAL 005336 GØ FIND DATA IN TABLE
005137/ 104  JMP 003333 GØ PRINT IT
005142/ 106  CAL 005063                INX,DCX RØUTINE
005145/ 362  LLC        SET UP ADR FØR 3 BYTE TABLE
005146/ 320  LCA        SAVE 'A' FØR NØW
005147/ 303  LAD        GET BINARY DATA
005150/ 044  NDI 001    MASK ALL BUT LS BIT
005152/ 104  JMP 005134 SEACH TABLE, GØ HØME
005155/ 362  LLC        GET ADR FØR 5 BYTE TABLE
005156/ 104  JMP 003330 GØ TØ ØUTPUT
005161/ 362  LLC        TABLE ADR     INP/ØUT/RST
005162/ 343  LED
005163/ 106  CAL 005305 Ø/P SYMBØL
005166/ 307  LAM        FETCH DATA
005167/ 044  NDI 300    CHECK BITS 6-7
005171/ 307  LAM        AND RESTØRE DATA
005172/ 150  JTZ 005210 JMP IF 00XXXXXX (RSI)
005175/ 044  NDI 076    MASK TØ 00XXXXX0
005177/ 012  RRC        SET UP 1/Ø PØRT NØ.
005200/ 066  LLI 352    PUT THE NUMBER AWAY
005202/ 056  LHI 013    FØR NØW
```

```
005204/ 370  LMA
005205/ 104  JMP 003364 GØ TØ ØUTPUT
005210/ 044  NDI 070    MASK DATA TØ 00XXX000 (RST NØ.)
005212/ 104  JMP 005200 GØ ØUTPUT IT
005215/ 362  LLC                        JMP/CAL/RET GRØUP
005216/ 056  LHI 004    SET UP FØR TABLE
005220/ 347  LEM        FETCH J,C,ØR R FRØM TABLE
005221/ 303  LAD        RESTØRE BINARY
005222/ 044  NDI 307    MASK AND CHECK IF UNCØNDITIØNAL
005224/ 060  INL        TRANSFER
005225/ 060  INL
005226/ 060  INL
005227/ 277  CPM
005230/ 150  JTZ 003330 YES-GØ TØ ØUTPUT
005233/ 066  LLI 210    NØ-LØØK UP CØNDITIØN
005235/ 303  LAD        IN 3 BYTE TABLE
005236/ 044  NDI 070    MASK ALL BUT CØNDITIØN
005240/ 106  CAL 005336 GØ SEARCH
005243/ 321  LCB        CHAR 3
005244/ 310  LBA        CHAR 2
005245/ 304  LAE        CHAR 1
005246/ 104  JMP 003333 GØ ØUTPUT
005251/ 303  LAD        GET DATA      RØTATE GRØUP
005252/ 242  NDC        MASK AS PER TABLE
005253/ 066  LLI 160    RØT IN 3 BYTE TABLE
005255/ 046  LEI 122    LØAD E WITH 'R' AND PRETEND
005257/ 104  JMP 005240 ITS A TRANSFER
005262/ 362  LLC        SET UP ADR    LØAD IMMEDIATE
005263/ 056  LHI 013
005265/ 076  LMI 111    LØAD 3RD CHAR AS 'I' (SØURCE REG.)
005267/ 104  JMP 005276 AND TREAT AS ØRDINARY LØAD
005272/ 303  LAD        GET DATA      LØAD (REG TØ REG)
005273/ 106  CAL 005067 GET SØURCE REG
005276/ 061  DCL
005277/ 061  DCL
005300/ 076  LMI 114    LØAD 'L' AS 1ST CHAR
005302/ 104  JMP 003336 GØ TØ ØUTPUT
005305/ 106  CAL 005352 3 BYTE TRANSFER
005310/ 106  CAL 005104 3 BYTE LØAD
005313/ 016  LBI 240                    PRINT 3 BYTES (RØUTINE)
005315/ 025  RST 020    Ø/P TWØ BLANKS
005316/ 025  RST 020
005317/ 066  LLI 350    ADR ØF FIRST CHAR
005321/ 056  LHI 013
005323/ 317  LBM        FETCH IT
005324/ 025  RST 020    PRINT
005325/ 060  INL        NEXT CHAR
005326/ 317  LBM
005327/ 025  RST 020    PRINT
005330/ 060  INL        ØNCE MØRE NØW
005331/ 317  LBM
005332/ 025  RST 020
005333/ 104  JMP 001023 GØ GET MØRE DATA
005336/ 056  LHI 004                    3 BYTE TABLE SEARCH
005340/ 277  CPM        CØMPARE
005341/ 150  JTZ 005356 EXIT IF FØUND
005344/ 060  INL        NEXT ENTRY
005345/ 060  INL
005346/ 060  INL
005347/ 104  JMP 005340 LØØP
005352/ 056  LHI 004                    3 BYTE TRANFER
005354/ 061  DCL
005355/ 327  LCM        GET 3RD CHAR FRØM TABLE
005356/ 061  DCL
005357/ 317  LBM        2ND CHAR
005360/ 061  DCL
005361/ 307  LAM        1ST CHAR
005362/ 007  RET        GØ!
005363/ 106  CAL 000362 INCR CLP      LINE CHECK
005366/ 066  LLI 353
005370/ 347  LEM        FETCH LINE CØUNT
005371/ 041  DCE        UPDATE IT
005372/ 150  JTZ 005000 END ØF PAGE Ø/P 3 LF'S
005375/ 104  JMP 005013 ØK GØ ØN TØ NEXT LINE
```

modular micros martin research

```
006000/ 302  LAC         GET 1ST CHAR  SYMBØLIC LØAD (LDS)
006001/ 074  CPI 114     IS IT AN 'L'
006003/ 110  JFZ 006122 NØ-TEST FØR 'R'
006006/ 016  LBI 306     PARTIAL WØRD  LØAD INSTRUCTIØN
006010/ 303  LAD         LØØK AT 2ND CHAR
006011/ 146  CAL 006345 ENCØDE AS DETINATIØN REG
006014/ 002  RLC
006015/ 002  RLC
006016/ 002  RLC
006017/ 201  ADB         STASH WITH PARTIAL WØRD
006020/ 310  LBA         IN 'B'
006021/ 006  LAI 111     IS 3RD CHAR AN 'I' ?
006023/ 274  CPE
006024/ 150  JTZ 006043 YES- GØ TØ IMMEDIATE RØUTINE
006027/ 304  LAE         NØ-ENCØDE SØURCE REGISTER AS ABØVE
006030/ 146  CAL 006345
006033/ 320  LCA
006034/ 301  LAB         GET DUMMY WØRD
006035/ 044  NDI 370     DISCARD BITS 0-3 (A=3X6)
006037/ 202  ADC         AND PUT IN THE REAL ØNE
006040/ 144  JMP 006046 NØW GØ CLEAN UP
006043/ 006  LAI 077                  IMMEDIATE LØAD
006045/ 241  NDB         MASK TØ 00XXXXXX
006046/ 340  LEA         A HAS INSTR   FINISH RØUTINE
006047/ 146  CAL 001023 GET CLP
006052/ 374  LME         PUT INSTR THERE
006053/ 300  LAA         NØP (NØT USED)
006054/ 146  CAL 006320 DECØDE LENGTH
006057/ 146  CAL 000311 INCR CLP
006062/ 012  RRC         CHECK LENGTH BITS
006063/ 100  JFC 003014 LEAVE US WHEN NØ MØRE BITS IN CARRY
006066/ 340  LEA         NØT DØNE-SAVE THE BITS
006067/ 016  LBI 240     PRINT A BLANK
006071/ 025  RST 020
006072/ 041  DCE         IS IT A 3BYTE INSTR?
006073/ 160  JTS 006105 SIGN FLAG TELLS ALL (SF=1 FØR 3 BYTE INSTR)
006076/ 106  CAL 001111 GET DATA AND INPUT
006101/ 250  XRA         CLEAR A
006102/ 104  JMP 006057 AND LØØP
006105/ 106  CAL 000311 INCR CLP      3BYTE (MUST WANT AN ADR)
006110/ 106  CAL 001023 GET MØRE DATA
006113/ 106  CAL 001212 AND STØRE TWØ BYTES (CLP,CLP-1)
006116/ 104  JMP 006101 GØ BACK TØ LØØP
006121/ 377  HLT         UNUSED HALT(1)
006122/ 074  CPI 122                  TEST FØR 1ST CHAR ='R'
006124/ 110  JFZ 006146 NØ- KEEP LØØKING
006127/ 026  LCI 005     IS IT A RØTATE?
006131/ 066  LLI 156
006133/ 106  CAL 003215 SEARCH 3 BYTE TABLE
006136/ 066  LLI 202
006140/ 150  JTZ 006234 IF NØ FIND,TEST FØR RETURN
006143/ 144  JMP 006046 GØ FINISH UP
006146/ 314  LBE                      ACC GRØUP
006147/ 343  LED         PUT CHARACTERS AWAY
006150/ 332  LDC
006151/ 026  LCI 013     SET UP TABLE SEARCH
006153/ 066  LLI 236     (ACC GRØUP,IN(R),DC(R))
006155/ 106  CAL 003215 SEARCH TABLE
006160/ 150  JTZ 006214 NØ FIND, KEEP LØØKING
006163/ 320  LCA         GET 1ST CHAR
006164/ 044  NDI 200     CHECK FØR IMMEDIATE INSTR
006166/ 302  LAC         RESTØRE CHAR
006167/ 341  LEB
006170/ 110  JFZ 006020 GØ AWAY IF IMMEDIATE INSTR
006173/ 301  LAB         TEST THE 3RD CHAR
006174/ 106  CAL 006345 ENCØDE AS A REGISTER
006177/ 002  RLC
006200/ 002  RLC
006201/ 002  RLC
006202/ 202  ADC         ADD TØ PARTIAL WØRD
006203/ 104  JMP 006046 FINISH UP
006206/ 106  CAL 001023 6 BYTES NØT USED (!)
006211/ 104  JMP 003143 (!!!)
006214/ 323  LCD                      TRANSFER GRØUP (JMP,CAL,RET)
```

```
006215/ 334  LDE          MUSICAL REGISTERS
006216/ 341  LEB
006217/ 066  LLI 172      START ØF TABLE (JMP)
006221/ 302  LAC
006222/ 277  CPM          TRY 1ST CHAR
006223/ 150  JTZ 006234   JUMP IF FIND
006226/ 066  LLI 176      TRY 'CAL'
006230/ 277  CPM
006231/ 110  JFZ 003131   NO (!) MUST BE AN ERRØR
006234/ 060  INL          IS IT UNCØNDITIØNAL?
006235/ 026  LCI 002
006237/ 146  CAL 003215   THEN GØ TØ FINISH
006242/ 110  JFZ 006046   GET PART WØRD
006245/ 061  DCL          AS MUCH AS WE CAN
006246/ 307  LAM          BLANK ØUT SØME
006247/ 044  NDI 303
006251/ 310  LBA          AND LØØK UP THE CØNDITIØN
006252/ 066  LLI 206
006254/ 026  LCI 011
006256/ 146  CAL 003215
006261/ 150  JTZ 003131   NØ-FIND ERRØR(!)
006264/ 201  ADB          ADD IN CØNDITIØN BITS
006265/ 144  JMP 006046   FINISH IT
006270/ 343  LED                          INP/ØUT/RST
006271/ 016  LBI 240      ENTER AS MØNITØR RØUTINE
006273/ 025  RST 020      PRINT A BLANK
006274/ 146  CAL 000205   INPUT THE ØCTAL ARGUMENT
006277/ 300  LAA          NØP (AGAIN -REALLY TØM !!)
006300/ 304  LAE          GET 2ND CHAR
006301/ 074  CPI 123      IS IT AN 'S'
006303/ 307  LAM          GET THE ØCTAL ARGUMENT
006304/ 046  LEI 005      ASSUME IT'S RST
006306/ 150  JTZ 006314   AND SKIP AHEAD IF IT IS
006311/ 002  RLC          MUST BE INP/ØUT-RØTATE ARGUMENT
006312/ 046  LEI 101      AND PUT THE REST INTØ E
006314/ 204  ADE          ADD THE TWØ PARTS TØGETHER
006315/ 144  JMP 006046   AND FINISH (GØØDBYE CRUEL WØRLD)
006320/ 250  XRA                        INSTRUCTIØN LENGTH TEST
006321/ 310  LBA          CLEAR REGISTERS
006322/ 304  LAE          GET DATA
006323/ 044  NDI 305
006325/ 074  CPI 004      IS IT IMMEDIATE?
006327/ 150  JTZ 006342   YES-BEGØNE
006332/ 044  NDI 301
006334/ 074  CPI 100      IS IT A TRANSFER?
006336/ 301  LAB          CLEAR A
006337/ 013  RFZ          PASS IF 3 BYTE (JMP,CAL GRØUP)
006340/ 010  INB          NØW SET UP B
006341/ 010  INB
006342/ 010  INB          CØME HERE IF 2 BYTE
006343/ 301  LAB          SØ NØW A IS 001 ØR 003
006344/ 007  RET          GØ HØME AND TELL ABØUT IT
006345/ 066  LLI 370                    REGISTER DECØDE
006347/ 056  LHI 006      LØØK AT TABLE
006351/ 277  CPM          TEST
006352/ 110  JFZ 006361   NØ FIND -LØØP
006355/ 306  LAL          A FIND! GET THE ADDRESS
006356/ 044  NDI 007      MASK 00000XXX
006360/ 007  RET          AND RETURN WITH A NUMBER
006361/ 060  INL          NEXT VALUE
006362/ 110  JFZ 006351   NØT ZERØ GØ LØØP
006365/ 104  JMP 003131   NØT IN TABLE- ITS AN ERRØR FØLKS
```

REGISTER LØØK UP TABLE

LØCATIØN	REGISTER	BINARY	ASCII
006370	A	0	101
006371	B	1	102
006372	C	2	103
006373	D	3	104
006374	E	4	105
006375	H	5	110
006376	L	6	114
006377	MEMØRY(M)	7	115

REV. A

This data sheet describes a simple program for testing random access memory installed in a *MIKE 2* microcomputer. Useful in testing both small and large blocks of memory, SP2210-4 can easily be entered into the *MIKE 2* using the Model 420 Console Board.

Equipment necessary for this software package include a *MIKE 2* system, based on the 8008 or 8008-1 microprocessor. The system must have a *MIKE 2* Monitor program installed, because SP2210-4 uses subroutines included in that PROM.

The program is typically entered into page 010 in memory, and can be used to test RAM at any other location. As the test proceeds, the locations of RAM under test flash rapidly on the Model 420 Console Board. If a bad location in RAM is found, the CPU is made to halt--indicated by a red HALT light on the CPU board--and the memory location at which the fault was found is displayed on the six Console Board digits. The same HALT condition occurs when the program runs out of RAM within the system. This makes for fast and easy diagnosis of memory problems in a *MIKE 2* microcomputer.

A 4K RAM test takes a little less than 10 seconds with a standard 8008 CPU--still less with an 8008-1.

TECHNICAL DATA

SEC. 1 PROGRAM DESCRIPTION

1.1 *Addressing* The operator enters a high-order memory address into the *MIKE 2* before executing this program. This allows the SP2210-4 to test RAM starting at location 000 of any page in memory. However, since the program itself is stored in RAM, the page where it is stored cannot itself be tested.

1.2 *First Test* The SP2210-4 makes two passes through RAM. During the first pass, the program stores into each location in RAM an eight-bit number derived from the sum of its H and L addresses. After writing that number into RAM, the program reads back from the same location, comparing the stored value with what was just stored there. If these values correspond, the program steps to the next location. This procedure continues until the program reaches a location where the comparison fails. Either a malfunctioning RAM chip is present, or no RAM has been installed at this location. The second pass of the RAM test then begins.

1.3 *Second Test* During the second pass, the program returns to the first page under test and reads each location, comparing the contents of each location--which were stored there during the first phase of the test--with what should be stored there, the H+L data. This time, if the stored data fails to compare with the H+L data, the program halts.

1.4 *Address Display* During both the first and second passes, the H/L address is loaded into the Console Board as each location in RAM is tested. The display flickers rapidly as testing proceeds. When a comparison fails during the second pass,

SEC. 1 PROGRAM DESCRIPTION (cont'd)

the program halts, and the Console displays the current memory location.

The left-hand three digits display the high-order memory address--or page number--in octal. The low-order address appears on the right-hand three digits.

SEC. 2 OPERATING INSTRUCTIONS

2.1 *Entering the Program* The SP2210-4 program is first entered into page 010 in RAM. A program listing with comments is shown on page 3 of this data sheet. Also shown is a simple dump of memory contents, showing the proper order for entering memory words.

After locations 010000 through 010040 have been programmed, the user should return to location 010000 and step through the program, to make sure it has been entered properly.

2.2 *Setting Up the Page Address* The user now sets up the page address of the RAM to be tested. This can be any page except for that on which the 2210-4 program itself has been entered. This value is entered on the Console Board keyboard, and loaded into the H register with the LOAD H key.

2.3 *Starting the Program* The user now depresses the EXECUTE key on the Console Board. This causes the computer to start executing the program, stored at location 010000. The RAM test proceeds as described above.

1975, Martin Research, Northbrook, IL 60062 USA. All rights reserved.

3	2	Rev	ECO	By	Ck	Date
A	A	A	016	KB	BR	12/9/75

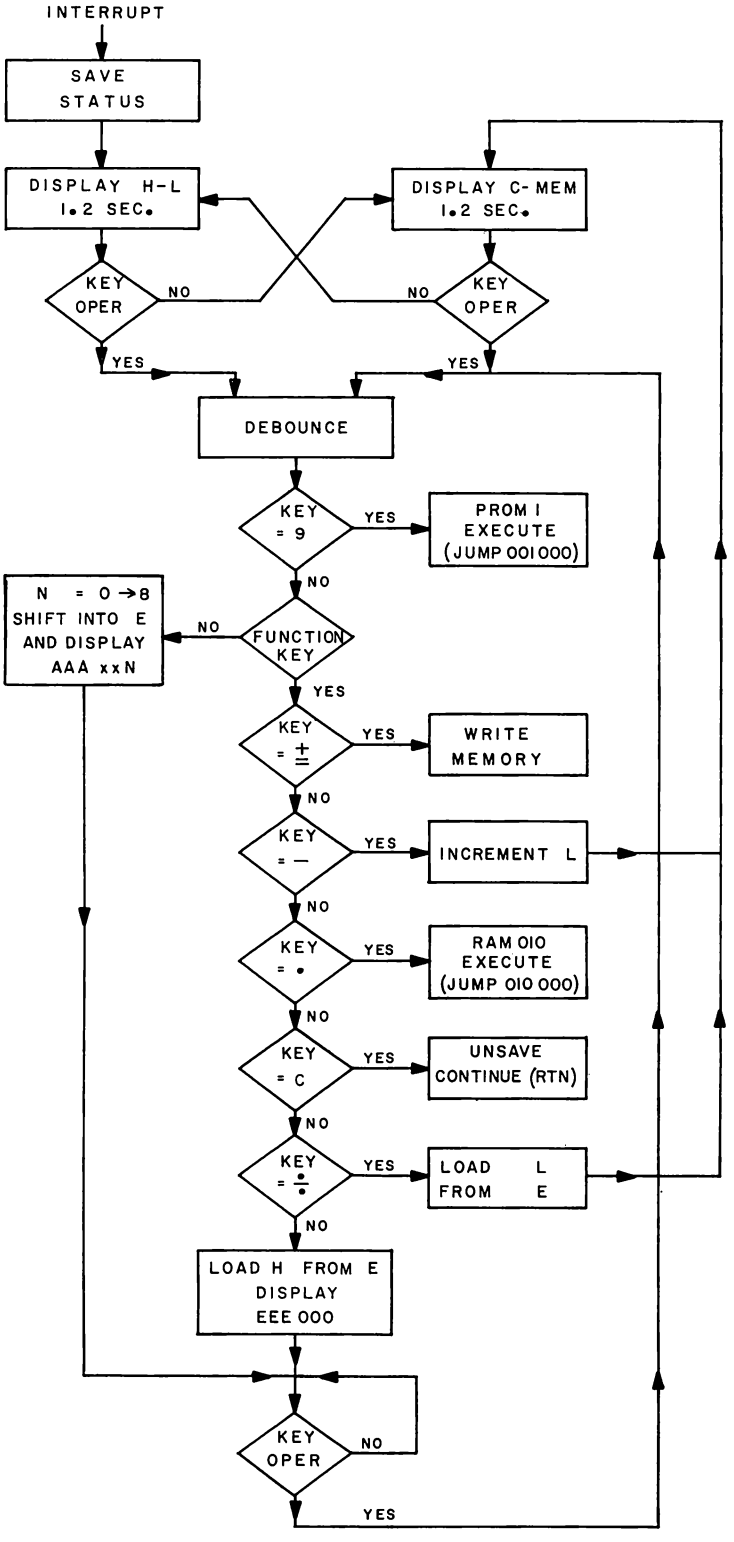
```
010000/ 051   DCH          DROP BACK FOR FIRST INCREMENT
010001/ 335   LDH          SAVE FOR SECOND PASS
010002/ 066   LLI 377      SET UP L REG
010004/ 106   CAL 000054   INCREMENT H AND L REGISTERS
010007/ 106   CAL 000322   DISPLAY IN OCTAL ON CNSL BD
010012/ 306   LAL          DATA IS L REGISTER--
010013/ 205   ADH          PLUS H REGISTER
010014/ 370   LMA          STORE IN RAM
010015/ 277   CPM          COMPARE IF IT IS RAM
010016/ 150   JTZ 010004   LOOP IF IT IS
010021/ 353   LHD          SECND PASS--GET START ADRS
010022/ 066   LLI 377      SET UP L AGAIN
010024/ 106   CAL 000054   INCREMENT H AND L
010027/ 106   CAL 000322   DISPLAY IN OCTAL ON CONSOLE
010032/ 306   LAL          DATA IS L REGISTER--
010033/ 205   ADH          PLUS H REGISTER
010034/ 277   CPM          COMPARE WITH RAM DATA
010035/ 150   JTZ 010024   LOOP IF OK
010040/ 000   HLT          HALT WITH FIRST BAD RAM LOCATION
```

```
010000/  051  335  066  377  106  054  000  106
010010/  322  000  306  205  370  277  150  004
010020/  010  353  066  377  106  054  000  106
010030/  322  000  306  205  277  150  024  010
010040/  000
```

NOTE: PROPRIETARY INFORMATION. NOT TO BE DISCLOSED WITHOUT EXPRESS WRITTEN PERMISSION.

DRAWN BY KB	CHECKED BY BR	DRAWING NO.
DATE 12/5/75	DATE 12,7,75	SP2210-4

A flow chart and software listings for the *MIKE 2* Monitor appear on this and the following pages. For an explanation on how the system operates, please see the *Operating Instructions.*



REV. A

```
;
;       MIKE2 MONITOR - VERSION 13
;       --------------------------
;
;
;
;       ONE PAGE MONITOR PROGRAM TO CONTROL KEYBOARD
;
        INTEL$                  . DEFINE 8008 ASSEMBLER STUFF
;
;       EQU'S FOR MONITOR PROGRAM
;
RAM     EQU     010*256         . RAM IS CURRENTLY PAGE 010
SAVE    EQU     013*256+0370    . SAVEAREA FOR STATUS ON INTERRUPTS
RAMPGM  EQU     RAM             . PROGRAM STARTS AT PAGE 'RAM'
PROM1   EQU     001*256         . NEXT PROM IS PAGE ONE
;
;       SUBROUTINES USEFUL TO OTHER PROGRAMS:
;
;       LOC     NAME            DESCRIPTION
;       ---     ----            -----------
;
;       001     UNKEY           WAIT UNTIL ALL KEYS UNPRESSED (DEBOUNCE)
;       010     RST010          TIMED LOOP OF 6,667 MSEC
;       020     RST020          PROGRAM BREAKPOINT
;       030     RST030          READ NEXT CHARACTER FROM CALC KEYBOARD
;       040     RST040          WAIT 1 SEC., OR A CALC KEY IS DEPRESSED
;       054     INHL            INCREMENT L (AND H) REGISTER
;       060     RST060          SEE IF ANY KEYS DEPRESSED (BUT DON'T READ)
;       073     STATUS          SAVE 8008 STATUS AND REGISTERS
;       151     REST            RESTORE SAVED 8008 STATUS AND REGISTERS
;       322     PARTHL          DISPLAY H AND L REGISTER IN OCTAL
;       350     PARTC           DISPLAY REGISTER C IN OCTAL (WITH FILLS)
;       363     EXP             EXPLODE REG C INTO OCTAL DIGITS (FOR OUTPUT)
;
;
;       MIKE2 - MONITOR FUNCTIONS
;       -------------------------
;
;       KEY     HEX             FUNCTION
;       ---     ---             --------
;
;       9       9               OFF TO PROM1
;
;       =       A               WRITE DISPLAY ACCUM INTO MEMORY
;                               CLEAR DISPLAY ACCUM
;                               INCREMENT L AND H REGISTERS
;                               DISPLAY CONTENTS OF MEMORY
;
;       -       B               INCREMENT L AND H REGISTERS
;                               CLEAR DISPLAY ACCUM
;                               DISPLAY CONTENTS OF MEMORY
;
;       *       C               LOAD H REGISTER FROM DISPLAY ACCUM
;
;       /       D               LOAD L REGISTER FROM DISPLAY ACCUM
;                               DISPLAY CONTENTS OF MEMORY
;
;       .       E               OFF TO RAM
;
;       C       F               CONTINUE WHERE INTERRUPTED
;                               (WITH UPDATED REGISTERS IF CHANGED)
;
;       0-7     0-7             INSERT DIGIT INTO DIPLAY ACCUM
;
```

© 1975 Martin Research, Northbrook, IL 60062 USA. All rights reserved.

8	7	6	5	4	3	2	Rev	ECO	By	Ck	Date
A	A	A	A	A	A	A	A	016	KB	BR	12/4/75

```
                .       REGISTER AND STATUS SAVEAREA:
                .       ----------------------------
                .
                .       PAGE    LOC             REGISTER
                .       ----    ---             --------
                .
                .       013     370             A
                .       013     371             B
                .       013     372             C
                .       013     373             D
                .       013     374             E
                .       013     375             STATUS
                .       013     376             H
                .       013     377             L


                .         ----------
                .
                .       LOCATION ZERO CONTAINS THE INTERRUPT JAM INSTRUCTION
                .
000000  025     RST000  RST     020             . OFF TO SAVE STATUS ON INTERRUPT
                .
                .       HERE TO WAIT UNTIL ALL KEYS 'UN' PRESSED
                .
000001  015     UNKEY   RST     010             . WAIT 6.667 MSEC
000002  065             RST     060             . ANY KEYS DEPRESSED?
000003  120             JFS     UNKEY           . LOOP IF ONE (OR MORE) KEYS DEPRESSED
000004  001
000005  000
000006  301             LAB                     . COPY 'B' INTO 'A' ALSO (SEE RST 030)
000007  015             RST     010             . WAIT 6.667 MSEC
                                                . MERGE TO WAIT ANOTHER 6.667 MS AND RETURN
                .
                .       LOOP TO WAIT 6.667 MSEC
                .
                        RES     010-$           . ALIGN FOR RST 010
000010  026     RST010  LCI     0147            . TIME CONSTANT OF 6.667 MSEC
000011  147
000012  021     TLOOP   DCC                     . DECREMENT COUNTER
000013  110             JFZ     TLOOP           . LOOP UNTIL COUNTER ZERO
000014  012
000015  000
000016  007             RET                     . RETURN
                .
                .       INTERRUPT OR PROGRAM BREAKPOINT COMES HERE
                .
000017                  RES     020-$           . ALIGN FOR RST 020
000020  106     RST020  CAL     STATUS          . SAVE CPU STATUS
000021  073
000022  000
000023  046             LEI     000             . CLEAR DISPLAY ACCUM REGISTER
000024  000
000025  104             JMP     DISPLY          . OFF TO MONITOR (DISPLAY MEMORY)
000026  261
000027  000
                .
                .       HERE TO READ NEXT KEY INTO REGISTERS 'A', AND 'B'
                .       DESTOYS A,B,C
                .
                        RES     030-$           . ALIGN FOR RST 030
000030  065     RST030  RST     060             . ANY KEYS DEPRESSED?
000031  160             JTS     RST030          . LOOP UNTIL KEY DEPRESSED
000032  030
000033  000
000034  312             LBC                     . COPY DEPRESSED KEY NUMBER
000035  104             JMP     UNKEY           . NOW MERGE TO WAIT FOR KEY TO BE 'UN' PRESS
000036  001
000037  000
```

```
                        .
                        .       HERE TO APPEAR TO PAUSE TO OPERATOR -->
                        .       UNLESS A KEY IS DEPRESSED
                        .
                                RES     040-$           . ALIGN FOR RST 040
000040  016     RST040  LBI     0200            . OUTSIDE TIME CONSTANT FOR 1.2 SEC
000041  200
                        .
                        .       INNER LOOP OF A FEW MILLESECONDS (6.667) OR UNTIL KEY DEPRESSED
                        .
000042  015     RST042  RST     010             . WAIT 6.667 MSEC
000043  065             RST     060             . ANY KEYS DEPRESSED?
000044  023             RFS                     . RETURN NOW IF SO.
000045  011             DCB                     . LOOP UNTIL
000046  120             JFS     RST042          . TIME CONSTANT = 0377
000047  042
000050  000
000051  007             RET                     . RETURN WITH CONDITION SIGN = 1
                        .
                        .       HERE TO INCREMENT H AND L REGISTER
                        .
000052  046     XINHL   LEI     000             . CLEAR E REG, MERGE INHL
000053  000
000054  060     INHL    INL                     . INCREMENT L REGISTER
000055  013             RFZ                     . RETURN IF NO OVERFLOW
000056  050             INH                     . INCREMENT H REGISTER
000057  007             RET                     . RETURN
                        .
                        .       SUBPROGRAM TO SEE IF ANY KEYS ARE DEPRESSED
                        .
                        .       CONDITION CODE SIGN=1 IF NO KEYS DEPRESSED
                        .       CONDITION CODE SIGN=0 IF ONE KEY DEPRESSED
                        .       REGISTER 'C' IS KEY DEPRESSED
                        .       DESTROYS A,B,C
                        .
                                RES     060-$           . ALIGN FOR RST 060
000060  026     RST060  LCI     020             . NUMBER OF KEYS TO CHECK
000061  020
                        .
                        .       HERE TO LOOK FOR A KEY DEPRESSED
                        .
000062  021     RST062  DCC                     . DECREMENT KEY LOOP COUNTER
000063  063             RTS                     . RETURN IF NO KEYS DEPRESSED
000064  302             LAC                     . COPY NEXT KEY NUMBER TO CHECK
000065  115             INP     006             . SEE IF THIS KEY IS DEPRESSED
000066  012             RRC                     . IS KEY DEPRESSED?
000067  003             RFC                     . RETURN IF KEY DEPRESSED
000070  104             JMP     RST062          . SEE IF MORE KEYS TO DO
000071  062
000072  000
                        .
                        .       SUBROUTINE TO SAVE 8008 STATUS AND REGISTERS
                        .
000073  113     STATUS  INP     005             . SAVE A REGISTER IN EXTERNAL 2 WORD QUEUE
000074  305             LAH                     . COPY H REGISTER
000075  113             INP     005             . SAVE H REGISTER IN EXTERNAL 2 WORD QUEUE
000076  306             LAL                     . COPY L REGISTER
000077  056     SAV     LHI     H(SAVE)         . GENERATE PAGE ADDRESS
000100  013
000101  066             LLI     L(SAVE+7)       . AND ENDING OFFSET OF SAVEAREA
000102  377
000103  370             LMA                     . SAVE L REGISTER
                        .
                        .       NOW SAVE STATUS BITS
                        .
000104  006             LAI     000             . ASSUME ZERO FLAG
000105  000
000106  150             JTZ     SAVC            . IF ZERO, THEN CHECK CARRY AND PARITY
000107  120
000110  000
000111  066             LAI     060             . ASSUME NO SIGN BIT
000112  060
000113  120             JFS     SAVC            . WE WERE RIGHT, SO CHECK CARRY AND PARITY
000114  120
000115  000
```

modular micros martin research

```
000116  006             LAI     0300            . SET SIGN BIT FLAG
000117  300
000120  032     SAVC    RAR                     . SAVE CARRY
000121  170             JTP     SAVW            . JUMP IF PARITY EVEN
000122  126
000123  000
000124  054             XRI     001             . SAVE PARITY ODD FLAG
000125  001
000126  061     SAVW    DCL                     . POINT TO NEXT SAVEAREA WORD
000127  113             INP     005             . SAVE STATUS, GET A , BUT WE WANT H
000130  113             INP     005             . SAVE A, GET H REGISTER
000131  370             LMA                     . SAVE H REGISTER IN MEMORY
000132  061             DCL                     . POINT TO NEXT SAVEAREA WORD
000133  113             INP     005             . GET STATUS
000134  370             LMA                     . SAVE STATUS
000135  061             DCL                     . POINT TO NEXT SAVEAREA WORD
000136  374             LME                     . SAVE E REGISTER
000137  061             DCL                     . POINT TO NEXT SAVEAREA WORD
000140  373             LMD                     . SAVE D REGISTER
000141  061             DCL                     . POINT TO NEXT SAVEAREA WORD
000142  372             LMC                     . SAVE C REGISTER
000143  061             DCL                     . POINT TO NEXT SAVEAREA WORD
000144  371             LMB                     . SAVE B REGISTER
000145  061             DCL                     . POINT TO NEXT SAVEAREA WORD
000146  113             INP     005             . RESTORE A REGISTER
000147  370             LMA                     . SAVE A REGISTER
000150  007             RET                     . HAVING SAVED EVERYTHING RETURN

                .
                .       SUBROUTINE TO RESTORE REGISTERS AND STATUS SAVED BY SAVE
                .
        151     PGMF    EQU     $               . 'F' --> CONTINUE WHERE INTERRUPTED
000151  056     REST    LHI     H(SAVE)         . GENERATE PAGE AND WORD
000152  013
000153  066             LLI     L(SAVE+7)       . OFFSET OF ENDING LOCATION OF SAVEAREA
000154  377
000155  307             LAM                     . GET L
000156  113             INP     005             . SAVE IT FOR NOW
000157  061             DCL                     . POINT TO NEXT SAVEAREA WORD
000160  307             LAM                     . GET H
000161  113             INP     005             . SAVE IT TOO, FOR NOW
000162  061             DCL                     . POINT TO NEXT SAVEAREA WORD
000163  307             LAM                     . RESTORE FLAG BITS
000164  061             DCL                     . POINT TO NEXT SAVEAREA WORD
000165  347             LEM                     . RESTORE E REGISTER
000166  061             DCL                     . POINT TO NEXT SAVEAREA WORD
000167  337             LDM                     . RESTORE D REGISTER
000170  061             DCL                     . POINT TO NEXT SAVEAREA WORD
000171  327             LCM                     . RESTORE C REGISTER
000172  061             DCL                     . POINT TO NEXT SAVEAREA WORD
000173  317             LBM                     . RESTORE B REGISTER
000174  061             DCL                     . POINT TO NEXT SAVEAREA WORD
000175  200             ADA                     . RESTORE FLAGS BITS
000176  307             LAM                     . GET A REGISTER
000177  113             INP     005             . SAVE A, GET L
000200  360             LLA                     . RETORE L REGISTER
000201  113             INP     005             . GET H REGISTER
000202  350             LHA                     . RETORE H REGISTER
000203  113             INP     005             . RESTORE A REGISTER
000204  007             RET                     . RETURN


                .
                .       A = WRITE NEXT LOCATION , THEN INC H AND L REGISTER
                .
000205  024     PGMA    SUI     013             . FORM 'B' ORIGIN
000206  013
000207  100             JFC     PGMB            . NOT 'A' SO TRY 'B'.
000210  214
000211  000
000212  374             LME                     . WRITE MEMORY FROM DISPLAY REGISTER
000213  250             XRA                     . CLEAR 'A' FOR MERGE WITH INC H AND L
```

```
                        .
                        .       B = INCREMENT H AND L REGISTER, THEN READ
                        .
000214  150     PGMB    JTZ     INCHL           . 'B' --> INCREMENT, THEN READ
000215  244
000216  000
                        .
                        .       E = JUMP TO RAM,  F = CONTINUE AFTER INTERRUPT
                        .
000217  024     PGME    SUI     003             . FORM 'E' ORIGIN
000220  003
000221  150             JTZ     RAMPGM          . 'E' --> OFF TO PROGRAM IN RAM
000222  000
000223  010
000224  100             JFC     PGMF            . 'F' --> CONTINUE LAST EXECUTION (RESTART)
000225  151
000226  000
                        .
                        .       D = LOAD THE 'L' REG, DISPLAY NEXT MEMORY LOCATION
                        .
000227  306     PGMD    LAL                     . COPY L REGISTER IN CASE WE ARE WRONG
000230  364             LLE                     . LOAD L REGISTER FROM DISPLAY ACCUM
000231  342             LEC                     . CLEAR DISPLAY ACCUM
000232  170             JTP     DISMEM          . 'D' --> LOAD 'L' AND DISPLAY MEMORY
000233  247
000234  000
                        .
                        .       C = LOAD THE 'H' REGISTER
                        .
000235  356     PGMC    LHL                     . LOAD 'H' REGISTER FROM OLD DISPLAY ACCUM
000236  360             LLA                     . RESTORE SAVED L REGISTER SINCE WRONG
000237  036             LDI     0356            . FILL CHARACTER
000240  356
000241  104             JMP     OUTACM          . MERGE TO OUTPUT ACCUM
000242  313
000243  000


                        .
                        .       HERE TO INC H AND L REGISTER
                        .
000244  106     INCHL   CAL     XINHL           . CLEAR E REG, INC H AND L REGISTERS
000245  052
000246  000
                        .
                        .       HERE TO DISPLAY CONTENTS OF MEMORY
                        .
000247  036     DISMEM  LDI     0314            . FILL CHARACTER OF 'CC' (FOR CONTENTS OF)
000250  314
000251  327             LCM                     . LOAD CONTENTS OF MEMORY
000252  106             CAL     PARTC           . OUTPUT C REGISTER
000253  350
000254  000
000255  045             RST     040             . PAUSE AND SEE IF ANOTHER KEY DEPRESSED
000256  120             JFS     FNC             . YES --> OFF TO PERFORM THAT FUNCTION
000257  270
000260  000
                        .
                        .       DISPLAY THE H AND L REGISTERS
                        .
000261  106     DISPLY  CAL     PARTHL          . DISPLAY 'H' AND 'L' REGISTER
000262  322
000263  000
000264  045             RST     040             . PAUSE AND SEE IF ANOTHER KEY DEPRESSED
000265  160             JTS     DISMEM          . NO --> DISPLAY CONTENTS OF MEMORY
000266  247
000267  000
                        .
                        .       NOTE FALL THRU TO NEXT PAGE
```

```
                        .
                        .       HERE TO GET NEXT FUNCTION
                        .
000270  035     FNC     RST     030             . READ THE NEXT KEY FROM KEYBOARD
000271  074             CPI     011             . THE DIGITS A - F ?
000272  011
000273  150             JTZ     PROM1           . '9' --> OFF TO PROM1
000274  000
000275  001
000276  100             JFC     PGMA            . YES THEN GO PROCESS SPECIAL CODES
000277  205
000300  000
                        .
                        .       HERE TO INSERT NEXT OCTAL DIGIT INTO DISPLAY ACCUM (REG E)
                        .
000301  304             LAE                     . COPY DISPLAY ACCUM
000302  002             RLC                     . SHIFT OVER
000303  002             RLC                     . THREE BITS
000304  002             RLC                     . FOR NEXT OCTAL DIGIT
000305  044             NDI     0370            . NOW CLEAR BITS FOR NEXT OCTAL DIGIT
000306  370
000307  201             ADB                     . PLUS THE NEXT CHARACTER
000310  340             LEA                     . STORE UPDATED LSR
000311  036             LDI     0252            . FILL CHARACTER 'AA' FOR ACCUM
000312  252
                        .
                        .       HERE TO OUTPUT DISPLAY ACCUM, AND FILL CHARACTER
                        .
000313  324     OUTACM  LCE                     . LEAST SIGNIFICANT PART
000314  106             CAL     PARTC           . OUTPUT THIS INFO
000315  350
000316  000
000317  104             JMP     FNC             . LOOK FOR NEXT FUNCTION
000320  270
000321  000


                        .
                        .       HERE TO OUTPUT 'H' AND 'L' REGISTER TO HEX DISPLAY PORTS
                        .
                        .       DESTROYS A,B,C
                        .
000322  326     PARTHL  LCL                     . LOAD LEAST SIGNIFICANT WORD
000323  106             CAL     EXP             . EXPLODE OCTAL DIGITS
000324  363
000325  000
000326  133             OUT     015             . OUTPUT RIGHT 2 DIGITS
000327  305             LAH                     . LOAD MOST SIGNIFICANT WORD
                        .
                        .       REGISTER A CONTAINS MOST SIGNIFICANT PART, AND
                        .       REGISTER 'B' THE REMAINDER FROM LEAST SIGNIFICANT PART
                        .
000330  012     PART2   RRC                     . MULTIPLY
000331  012             RRC                     . BY 2**3
000332  012             RRC                     . TO MOVE 3 BITS
000333  320             LCA                     . SAVE FOR H REGISTER
000334  012             RRC                     . ONE MORE BIT
000335  044             NDI     0160            . EXTRACT -210----
000336  160
000337  261             ORB                     . FORM   -210--76
000340  135             OUT     016             . OUTPUT H DIGIT AND REST OF L DIGITS
000341  106             CAL     EXP             . EXPLODE MODIFIED H REGISTER
000342  363
000343  000
000344  044             NDI     077             . KILL HIGH ORDER BITS
000345  077
000346  137             OUT     017             . OUTPUT THE HIGH 2 H DIGITS
000347  007             RET                     . RETURN
```

```
                        .
                        .       HERE TO OUTPUT BINARY REGISTER 'C' AND FILL CHARS IN REG 'D'
                        .
000350  106     PARTC   CAL     EXP             . EXPLODE BINARY REGISTER C
000351  363
000352  000
000353  133             OUT     015             . OUTPUT RIGHT MOST TWO DIGITS
000354  303             LAD                     . COPY FILL CHARACTERS
000355  137             OUT     017             . OUTPUT FILL CHARACTERS
000356  044             NDI     0360            . CLEAR RIGHT DIGIT
000357  360
000360  261             ORB                     . FORM FILL CHAR, OCTAL DIGIT
000361  135             OUT     016             . OUTPUT IT
000362  007             RET                     . RETURN

                        .
                        .       EXPLODE BINARY REG C INTO PRINTABLE OCTAL FOR HEX OUTPUT PORTS
                        .
                        .       DESTROYS A,B
                        .
                        .       NOTE THAT '-' DENOTES ZERO BITS
                        .
000363  302     EXP     LAC                     . COPY BINARY REGISTER
000364  002             RLC                     . POSITION BITS
000365  002             RLC                     . 7 AND 6
000366  044             NDI     003             . FORM ------76
000367  003
000370  310             LBA                     . SAVE IT
000371  302             LAC                     . COPY BINARY REGISTER
000372  044             NDI     0370            . ISOLATE XX543---
000373  370
000374  202             ADC                     . FORM X543-210
000375  044             NDI     0177            . FORM -543-210
000376  177
000377  007             RET                     . RETURN
```